DOCUMENT RESUME

ED 127 969 IR 003 934

AUTHOR	Hillelsohn, Michael J.
TITLE	Project IMPACT Courseware Subsystem: Volume I: Innovative Procedures for Development and Administration. Technical Report 74-1.
INSTITUTION	Human Resources Research Organization, Alexandria, Va.
SPONS AGENCY	Army Research Inst. for the Behavioral and Social Sciences, Arlington, Va.
REPORT NO	HumRRO-TR-74-1
PUB DATE	Feb 74
CONTRACT	DAHC-19-73-C-0004
NOTE	144p.; For a related document see IR 003 935
EDRS PRICE	MF-$0.83 HC-$7.35 Plus Postage.
DESCRIPTORS	Autoinstructional Aids; *Computer Assisted Instruction; Computer Oriented Programs; *Computer Programs; Computers; Computer Science Education; *Experimental Programs; Instructional Materials; *Instructional Systems; Instructional Technology; On Line Systems; Programing
IDENTIFIERS	COBOL; COmmon Business Oriented Language; Human Resources Research Organization; HumRRO; Instructional Decision Model; *Interactive Computer Systems; Learner Controlled Instruction

ABSTRACT

Instructional design, instructional documentation, and instructional materials are used in the Project IMPACT computer assisted instruction (CAI) system. Separation of course content and logic in the system permits individualized training controlled in part by the student and in part by an instructional decision model. The entire instructional system includes equipment, computer programing, management programing, and personnel. Examples of course materials drawn from the revised COBOL course (COBOL2) and the revised orientation course are included in the appendices, along with information on additional materials stored in the courseware files of the Project IMPACT system. (CH)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

HumRRO
Technical
Report
74-1

ED127969

Project IMPACT Courseware Subsystem: Volume I—Innovative Procedures for Development and Administration

Michael J. Hillelsohn

HumRRO Division No. 1 (System Operations)
Alexandria, Virginia

HUMAN RESOURCES RESEARCH ORGANIZATION

Work Unit: IMPACT-CATALYST

February 1974

IR 003 934

Approved for
public release;
distribution
unlimited.

Prepared for

U.S. Army Research Institute for the
Behavioral and Social Sciences
1300 Wilson Boulevard
Arlington, Virginia 22209

U.S. DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

The Human Resources Research Organization (HumRRO) is a nonprofit corporation established in 1969 to conduct research in the field of training and education. It is a continuation of The George Washington University Human Resources Research Office. HumRRO's general purpose is to improve human performance, particularly in organizational settings, through behavioral and social science research, development, and consultation. HumRRO's mission in work performed under contract with the Department of the Army is to conduct research in the fields of training, motivation, and leadership.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Published
February 1974
by
HUMAN RESOURCES RESEARCH ORGANIZATION
300 North Washington Street
Alexandria, Virginia 22314

Distributed under the authority of the
Chief of Research and Development
Department of the Army
Washington, D.C. 20310

FOREWORD

This report describes the instructional design, documentation, and materials of computer-based courseware developed in Project IMPACT. It is the first of two volumes to be issued about the courseware subsystem of this project.

Work Unit IMPACT is an advanced development project undertaken by the Human Resources Research Organization to provide the Department of the Army with a Computer-Administered Instruction (CAI) system. The National Science Foundation is also sponsoring HumRRO research on Instructional Decision Models (IDMs), with additional support provided by the James McKeen Cattell Fund.

The research is being conducted at HumRRO Division No. 1 (System Operations). Alexandria, Virginia. Dr. J. Daniel Lyons is the Director of the Division, and Dr. Robert J. Seidel is the Program Director. The innovations documented in this report were designed, developed, and tested over a period of several years. Many people contributed to their creation and operationalization, including (in alphabetical order): Ms. Lala J. Curry, who, in addition to her technical assistance, edited the contents of the report; Ms. Nancy Hibbits; Mr. Michael J. Hillelsohn; Ms. Caroline J. Huber; Dr. Harold Hunter; Dr. Felix F. Kopstein; Ms. Myrna Leibson; Mr. Richard D. Rosenblatt; Dr. Edward W. Schneider; Mr. Russel E. Schulz; Mr. Michael A. Smith; and Mr. William G. Underhill.

A complete list of publications issued to date under Project IMPACT is provided in the Bibliography of this report.

The identification of specific products is intended for research documentation purposes only, and does not constitute an official endorsement by HumRRO, the Department of the Army, the National Science Foundation, or the James McKeen Cattell Fund.

The HumRRO research begun under Work Unit IMPACT and continued under Work Unit CATALYST is conducted for the Department of the Army under Contract DAHC 19-73-C-0004. Computer-Administered Instruction research is conducted under Army Project 2Q063101D734. Research on Instructional Decision Models being conducted under National Science Foundation sponsorship is funded under Grant GJ-774, with additional funds from the James McKeen Cattell Fund.

Meredith P. Crawford
President
Human Resources Research Organization

BACKGROUND

The objective of Project IMPACT was to evolve a series of prototype systems of Computer-Administered Instruction (CAI) in order to produce a total CAI system that is effective, efficient, and cost-effective for operational use in a training/instructional environment.

The total prototype system includes four main components:

(1) Hardware—the computer, student stations, and related equipment.

(2) Software—the computer programming systems that control operation of the hardware.

(3) Courses of Instruction—the actual content and logic of courses administered by the computer.

(4) Instructional Decision Models (IDMs)—the rules and strategies by which specific course content is provided to an individual student.

The way in which the content of instruction is prepared, stored in the computer, managed by the computer while students are taking a course, and managed off-line by course authors and administrators is the key to the efficiency and effectiveness of the total CAI system. All components of the system interact in the courseware subsystem, where one visible product, a course of instruction, interfaces with the learner. Development of the instruction within the constraints specified by the other three subsystems, and administration of the instruction so that all facets of the total system are optimally employed for the ultimate benefit of the student are the subject of this report.

OBJECTIVES

The original Project IMPACT objective (in 1967) for developing a course of instruction was to provide a vehicle enabling validation of the components of the total CAI system. After the first iteration of the course was tested, a period of reassessment resulted in more stringent and exhaustive objectives for the courseware subsystem.

Maintaining the course materials as a valid vehicle for instructional design remained the primary objective. Additionally, the subsystem had become so large (one course was over 30 hours long) that efficient management techniques had to be incorporated—both on-line and off-line—for the development and administrative activities related to the courseware subsystem. Increased cost-effectiveness in the production of CAI materials was another goal during the second iteration of the courseware subsystem.

Also of extreme importance in the second iteration was the objective of flexibility of courseware. In individualized instruction, a teacher/author must be free to develop creative and adaptive materials to meet the student's needs as they vary over time throughout a course of instruction. Flexibility also implies that the courseware is amenable to change without disrupting the entire course administration.

APPROACH AND DEVELOPMENT

All of the objectives were met by developing an innovative set of standardized procedures for the design, production, and documentation of the IMPACT Courseware Subsystem.

v

The most important software development, using the computer as an efficient tool, was the physical and logical separation of instructional text from the instructional logic files. This Project IMPACT innovation aided the attainment of the aforementioned objectives by permitting on-line creation and modification of text and logic independently.

This capability resulted in effective file management, economy of text storage, reduced on-line development time, efficient on- and off-line retrieval and modification of each of the files, and more adaptive Instructional Decision Models (IDMs). Off-line developments, such as the File Activity Control System (FACS) and title listings, were also made possible and practical because of the text and logic separation. These products in turn made it feasible for course managers and authors/teachers to do much of their work off-line and thereby use relatively expensive computer time most effectively and efficiently.

Development of the subsystem was evolutionary and iterative. The requirements of potential users were anticipated, and a multidisciplinary team designed and implemented solutions to these requirements. After implementation, a period of assessment occurred, during which the innovation was evaluated. If successful, the technique became part of the subsystem; if not, additional solutions were tested until a satisfactory one was found.

CONTENTS

Chapter Page

1 **Introduction** 3
  Background of the Project 3
  Cycle I of the Project 3
  Cycle II of the Project 5
  Major Course Revision 5
  Organization of the Report 6

2 **Instructional Design** 7
  Instructional Strategies 7
  Instructional Objectives 9
  Course Structure 10
  Instructional Decision Model (IDM) Features 12
  Instructional Treatment 16
  Hardware/Software Considerations 18
    Input/Output Terminal 18
    Visual System 20
  Auxiliary Course Materials 21
  Staff Considerations 21

3 **Instructional Documentation** 23
  System Requirements 23
  Response Analysis Documentation 23
  Titles and Their Uses 25
    CRT Text 25
    Visuals 27
    Glossary Terms and Diagnostics 29
    Student Reference Manual 30
    FACS and MINI-FACS 30

4 **Instructional Materials** 32
  Text 32
  Visuals 36
  Glossary 36
  Student Reference Manual 37
  Supplies 37

5 **Implementation and Outcomes** 38

Bibliography 41

Appendices

A. Student Tasks in the COBOL1 Course 45
B. Description of the INTRO Course 49
C. Description and Objectives of the COBOL2 Course 52

Appendices *(Cont'd)*

Page
D. COBOL Programs Used in COBOL2 67
E. Sample Materials from the INTRO and COBOL2 Courses 86
F. Mnemonics and Messages 122
G. Listing of Glossary Terms and Diagnostics 127
H. Contents of the Student Reference Manual 133
I. SUPER-EDITOR 136
J. VOYEUR 139

Figures

1 Project IMPACT—Development Activities During Cycle I 4
2 Real-Time Software Components 8
3 Hierarchy of Objectives 9
4 Course Structure—Divisions, Modules, and Sections 11
5 Sample Page Groupings Within Instructional Units of a Section 13
6 Sample Page Components 14
7 Flowchart of Instructional Decision Model (IDM) Features Under Learner Control 15
8 Choicepoint in Course ROUTE 17
9 CRT Display Terminal 19
10 CRT Keyboard 19
11 Excerpt from a Full Title Listing for Text and Visuals 28
12 P-Scope Film 29
13 Off-line Software Components 31
C-1 COBOL2 Prerequisite Structure 57
C-2 Partial Path Diagram for Division B 58
I-1 Mode Element 137
J-1 Portion of VOYEUR Output 139
J-2 VOYEUR Element for Trouble Reports 140

Tables

B-1 Variations on Module IKB 51
I-1 SUPER-EDITOR Modes 136

Chapter 1

INTRODUCTION

The purpose of this report is to describe and explain a set of course materials (courseware) in COBOL Programming produced for computer-based instruction in a research and development project.

Related facets of the total instructional system are presented only in enough detail to provide a meaningful context for the development and implementation of the course itself. These facets include the interdependent subsystems of equipment (hardware), computer programming (software), and management programming (Instructional Decision Model and data analysis), as well as the learners and staff personnel. The history of this work in individualized computer-based learning materials is described, along with a brief discussion of project administration and sponsorship.

BACKGROUND OF THE PROJECT

Project IMPACT was conceived in late 1966 in a Technical Development Plan prepared by Robert J. Seidel and other staff members of HumRRO.[1] The research arm of the Army, a major sponsor of Project IMPACT, was very much interested in exploring the potential of the still relatively young technology of computerized training. COBOL Programming was selected as the first subject-matter vehicle for the development of a completely new approach to meeting critical training needs.

The project was launched in 1967 as a five-year advanced development effort to provide the Army with the specific requirements for a prototype system ready to put into operation. The original plan also included several programs of instruction dealing with a variety of subject matter of critical importance, as well as families of decision models of the instructional process (IDMs). Subsequent shifts in sponsor priorities resulted in some modification of original goals. Development of system capabilities evolved as the primary focus of the project, with Introduction to COBOL Programming as the major course vehicle.

CYCLE I OF THE PROJECT

Following is an excerpt from a status report of the project as of June 1969:[2]

> **Project IMPACT is a total systems development effort designed to provide . . . an effective, efficient, and economical computer-administered instruction (CAI) system. The acronym stands for Instructional Model Prototypes Attainable in Computerized Training. The objective of the research is to (a) develop two generations of prototype CAI systems with (b) accompanying prototype multi-path branching), individualized programs of instruction.**

---

[1] "Project IMPACT—Instructional Model Prototypes Attainable in Computerized Training," Technical Development Plan, Human Resources Research Office for the Department of the Army, December 1966.

[2] Robert J. Seidel *et al.*, *Project IMPACT: Description of Learning and Prescription for Instruction*, HumRRO Professional Paper 22-69, Human Resources Research Office, June 1969.

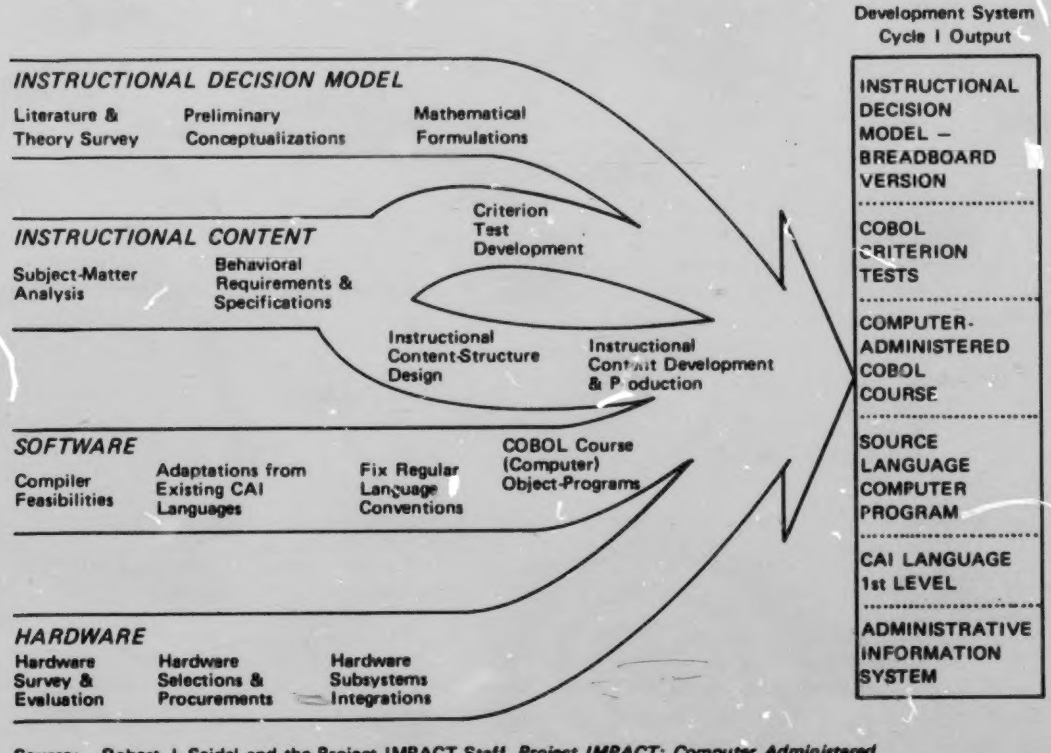
The system of instruction is to be capable of adapting to the capabilities, at the moment, of each individual trainee. This adaptiveness will be based both on the "entry characteristics" of the trainee and on his long-term and immediate response patterns within the course, so that each step in the instruction will be fitted directly to his needs at that point in the process. The instruction will also be made directly relevant to his specific job requirements.

An integrated, multi-faceted, and interdisciplinary approach is being used in the CAI developmental plan, which involves four phased development cycles. The development activities during each of the cycles are grouped under four major classifications: Hardware, Software, Instructional Content, and the Instructional Decision Model.

Work in Cycle I was undertaken at the beginning of FY 1968. Completion of Cycle I is forecast for the end of FY 1970. The products of Cycle I are shown in Figure 1.

Work in Cycle I is essentially devoted to the assembly of a "breadboard" version of the initial CAI system. Developmental activities on the Instructional Decision Model, the Instructional Content, the Hardware, and the Software components of the system are under way, and the initial versions of the various products are being provisionally integrated and adjusted to each other. These activities have been facilitated because of the subject matter expertise gained by the Human Resources Research Office in previous research under Work Unit METHOD which dealt with development of a linear programed course in fundamentals of computer programming. The nature and interrelationships of the various activities in Cycle I are also illustrated in Figure 1.

Project IMPACT—Development Activities During Cycle I



Development System
Cycle I Output
INSTRUCTIONAL DECISION MODEL
Literature & Theory Survey
Preliminary Conceptualizations
Mathematical Formulations
INSTRUCTIONAL CONTENT
Subject-Matter Analysis
Behavioral Requirements & Specifications
Criterion Test Development
Instructional Content-Structure Design
Instructional Content Development & Production
SOFTWARE
Compiler Feasibilities
Adaptations from Existing CAI Languages
Fix Regular Language Conventions
COBOL Course (Computer) Object-Programs
HARDWARE
Hardware Survey & Evaluation
Hardware Selections & Procurements
Hardware Subsystems Integrations
INSTRUCTIONAL DECISION MODEL – BREADBOARD VERSION
COBOL CRITERION TESTS
COMPUTER-ADMINISTERED COBOL COURSE
SOURCE LANGUAGE COMPUTER PROGRAM
CAI LANGUAGE 1st LEVEL
ADMINISTRATIVE INFORMATION SYSTEM

Source: Robert J. Seidel and the Project IMPACT Staff, ***Project IMPACT: Computer Administered Instruction Concepts and Initial Development,*** HumRRO Technical Report 69-3, Human Resources Research Office, March 1969.

Figure 1

> Cycle II involves a test of the "breadboard" version of the CAI system developed in Cycle I, to determine its effectiveness in training. On the basis of evaluation experience, this version will then be revised into the first-generation prototype of an operationally implementable CAI system.

Developmental activities in Instructional Content are summarized[1] as follows:

> (1) After establishment of course objectives, the director of Instructional Programming trained the staff in COBOL.
> (2) Each instructional programer developed, on the basis of his COBOL learning experience, a technique for teaching COBOL in a tutorial setting. Each instructional programer taught at least one student.
> (3) On the basis of the tutorials, the Instructional Programing team developed an outline of the basic elements of COBOL, establishing the behavioral objectives required for writing sample COBOL programs.
> (4) On the basis of the tutorials, the Instructional Programing team developed an outline for the content and organization of the Preliminary COBOL Course.
> (5) The Preliminary COBOL Course was prepared. Tutorial students were used at each rough-draft stage as an aid in refining the content.
> (6) Independently, criterion tests were written for the various levels of the course. The preliminary COBOL course was tested off-line with 12 high school and college students during July and August 1968.

CYCLE II OF THE PROJECT

During the second cycle of the project, the formative development and evaluation of the COBOL course concentrated on the refinement of Cycle I materials. Hard-copy versions of the glossary and visuals were made available on-line via the CRT and the Perceptoscope Visual System. An off-line session was added to the end of the course for student debugging of the final criterion program. A Student Reference Manual was created for this debugging session, along with simplified versions of COBOL Compiler diagnostics in the form of a manual card index file.

Appendix A gives brief descriptions of the tasks performed by students in the 18 on-line modules of the refined course (COBOL1).

The COBOL1 course was administered first to high school students for debugging and then to Army students for evaluation of the system. Data were collected and analyzed on a total of 42 students. The 12 high school students were recruited from nearby distributive education programs and the 30 Army students came from Project Transition.

MAJOR COURSE REVISION

During Cycle II of the Project, the need for rethinking and greater revision became apparent.

Evaluative data showed that smaller subsets of instruction were required for both learning and research purposes. These modular pieces of instruction were delineated by well-defined objectives and criterion tasks. Once the objectives were identified, an appropriate prerequisite structure was defined, with allowance made for more adaptive instruction than was previously available. The new prerequisite structure and selective refinement of the Entry Characteristics Test Battery precluded the need for giving subject matter pretests to the student groups.

---

[1] Robert J. Seidel and the Project IMPACT Staff, *Project IMPACT: Computer-Administered Instruction Concepts and Initial Development*, HumRRO Technical Report 69-3, Human Resources Research Office, March 1969.

The Instructional Decision Model (IDM) also underwent changes, not the least of which was the separation of the logic of the IDM from that of the instruction. The ramifications of this development were enormous because the IDM could be dynamic and unconstrained by the course logic. In fact, with this new flexibility, the converse became true and the IDM justifiably dictated constraints to the courseware.

The primary thrust of the course revision (COBOL2) was to produce a more viable vehicle for research on instructional strategies. However, a concomitant goal was to develop and test *in situ* the much-needed managerial tools and techniques that could be applied to making computerized training more efficient and economical.

ORGANIZATION OF THE REPORT

This report focuses on the innovative tools and techniques that were developed in Project IMPACT to cope with the myriad interactions between courseware and the other subsystems of hardware, software, IDM, and users. Further details about these subsystems are available in other Project IMPACT publications listed in the Bibliography at the end of this document.

The discussion of courseware is presented in the three major sections on Instructional Design, Documentation, and Materials. Examples are drawn from the revised COBOL course (COBOL2) and the revised orientation course (INTRO) which introduces the new student to the training situation. In order to meet the needs of a diverse group of readers, these samples are presented in the appendices, along with many other courseware details, perhaps of more interest to the specialist than the generalist.

The appendices are organized and sequenced for readers who might wish to peruse some details without continually referring to the body of the report. For the most part, Appendices A through D show the development of the instructional design, while E and F are oriented toward documentation. Appendices G and H indicate the subject matter coverage of on-line and off-line materials available to the student as instructional aids. Appendices I and J present brief descriptions of additional material stored in the courseware files of the Project IMPACT system.

Chapter 2

INSTRUCTIONAL DESIGN

Computer-Administered Instruction (CAI) developed in Project IMPACT was characterized by separate storage and retrieval of instructional content and logic, as shown in Figure 2. These files were interfaced through a combination time-sharing monitor and data-base management executive subsystem (called ZEUS). This is the first application of such a system that proved to be successful under operational conditions. The original proposal for the project was, indeed, a pioneer effort in attempting to design and develop a total system of computerized training based on this concept.

The separation of instructional text from instructional logic has many advantages over a system in which the instructional text is embedded within the instructional logic. One of the most important benefits is the increased cost-effectiveness that can result from several people being able to author the same piece of instruction simultaneously, thereby amortizing computer costs over more users and reducing total time on-line necessary to create a given segment of instruction. Economical use of file space is another benefit of this system, because the same piece of text can be retrieved for display throughout a course, but only one copy need be stored on the instructional text files.

This design also facilitates maintenance of the stored material, since the textual component can often be modified without causing major reformulations in the logic programs, and vice versa. Separate storage has also enabled a simple author language and authoring aids to be developed, allowing relatively inexperienced programming personnel to input most of the text on-line.

One example of the authoring aids developed in Project IMPACT is the program called SUPER-EDITOR (see Appendix I). In order to be optimally effective, this program was created as a "course" so that the authors could use it by "signing on" with student registration numbers. Therefore, the text was stored in the instructional text files and the logic in the instructional logic files.

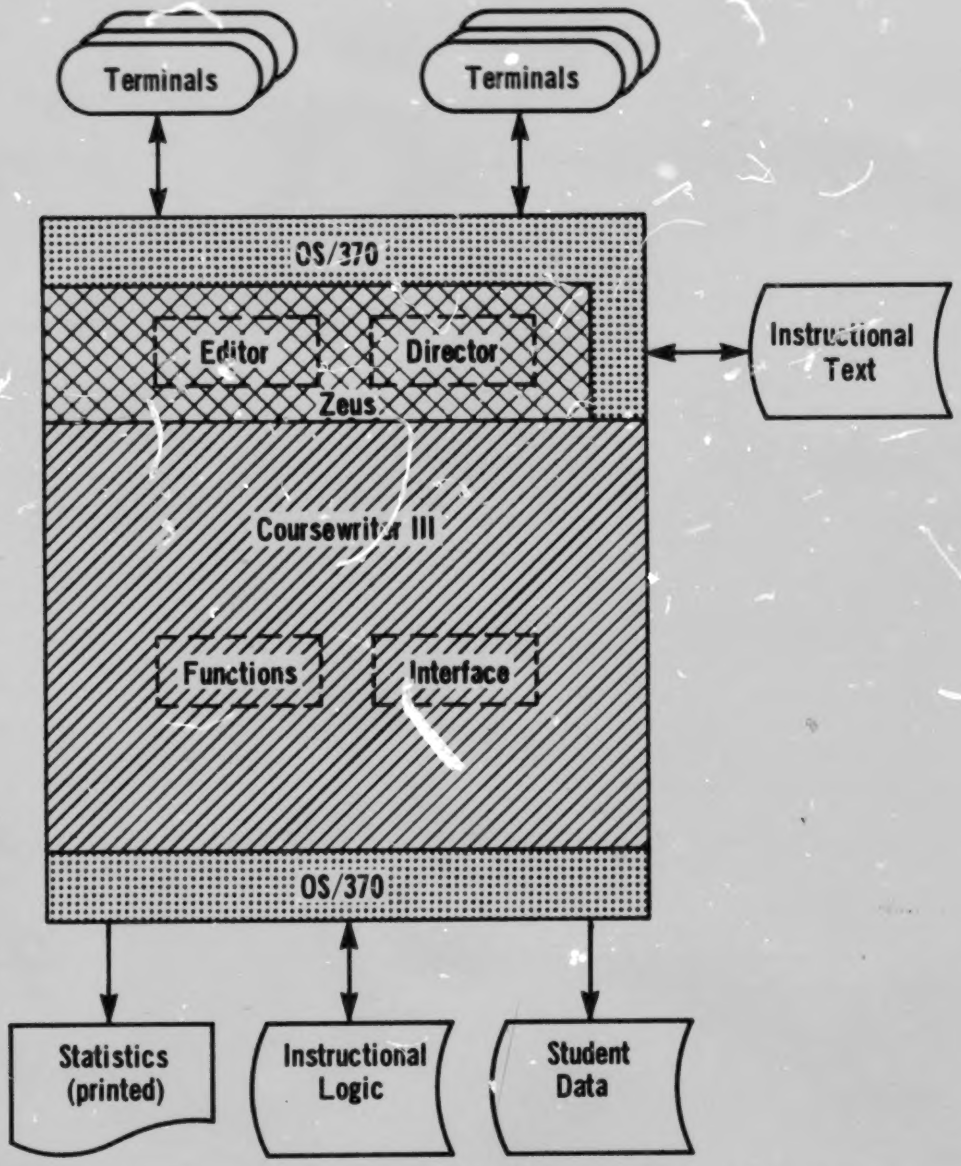
A similar "course" was created as an on-line proctoring aid for instructional administration. A description of this VOYEUR program is included in this report as Appendix J.

The ability to use "courses" in a variety of ways is clearly an advantage, since authors, administrators, and students can all work on the system at the same time. In Project IMPACT, the hardware/software subsystem allowed as many courses to be reviewed simultaneously as there were terminals in operation.

INSTRUCTIONAL STRATEGIES

In order to understand the features, including the on-line documentation that comprise an instructional package, one must be aware of the instructional and logical strategies of the courses. The pedagogical strategies and instructional design of the course define a system of rules for decision making which are labeled the Instructional Decision Model (IDM). The IDM is put into practice through the Interface (Figure 2). Here the instructional text, instructional logic, and the student (via his terminal) must interact, and each is constrained by the design elements of the IDM and the logical constraints of the Interface.

Real-Time Software Components


Terminals
Terminals
OS/370
Editor
Director
Zeus
Instructional Text
Coursewriter III
Functions
Interface
OS/370
Statistics (printed)
Instructional Logic
Student Data

Source: John Stelzer and Leslie Willis, *Project IMPACT Software Documentation: III. IMPACT's Data Evaluation System—Version I (IDES-I)*, HumRRO Research Product RP-D1-72-2, August 1972.

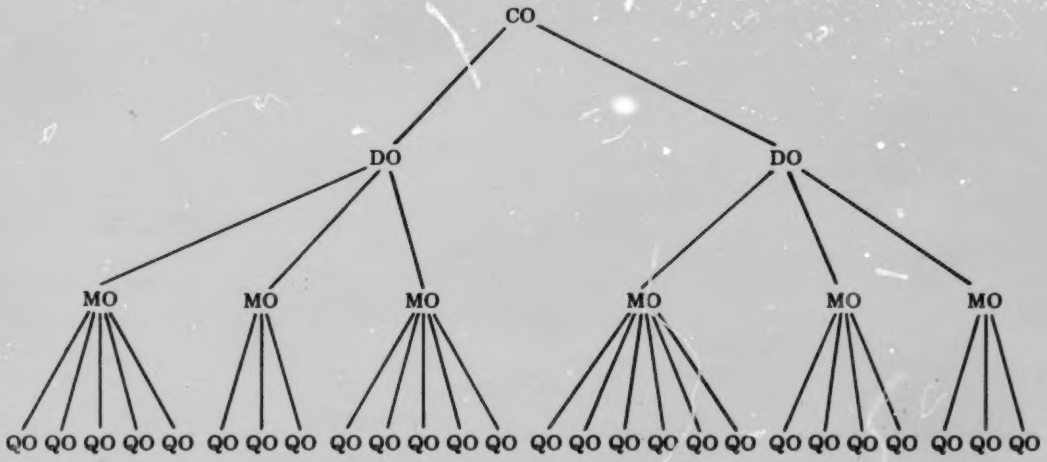
Figure 2

To the student these constraints appear quite acceptable, if they are perceived as constraints at all. They deal mainly with the ordering of the presentation of the instructional material, since it is the IDM, through the Interface, that can access and present any piece of text in the instructional files at any time. Within the IDM the rules for presentation are defined and the Interface carries out these rules. Since the whole system is based on rules and the practical hardware/software limitations relevant to their execution, the course author must structure his material so that it is compatible with both the rules and the limits of execution.

INSTRUCTIONAL OBJECTIVES

The first constraint that an author must consider is the established hierarchy[1] of objectives, shown in Figure 3.[2] His material must be structured so that it fits this design, wherein the course objective defines the singular terminal behavior for the course and the question objectives are single enabling objectives. Module (or topic) and division objectives are of an intermediate type, whereby they are of an enabling nature to whatever is above them (see Figure 3) and terminal to whatever is below them. There can only be one course objective. However, since the number of prerequisite skills will vary from

Hierarchy of Objectives


CO
DO
DO
MO
MO
MO
MO
MO
MO
QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO QO

Where: CO = Course objective
DO = Division objective
MO = Module objective
QO = Question objective

Figure 3

[1] "Hierarchy" is not theory-bound. As used in Project IMPACT, it refers to the ordering of input and output relationships among objectives in the course. Thus, it is not specified that a module requiring a chain of discriminatory responses precede a module requiring a principal behavior.

[2] The figure is purely schematic and does not represent any particular course.

course to course, there can be as many divisions, modules, and questions as required to teach the material. Because the module objective is the lowest-order terminal objective, it becomes the logical unit of instruction for an author.

COURSE STRUCTURE

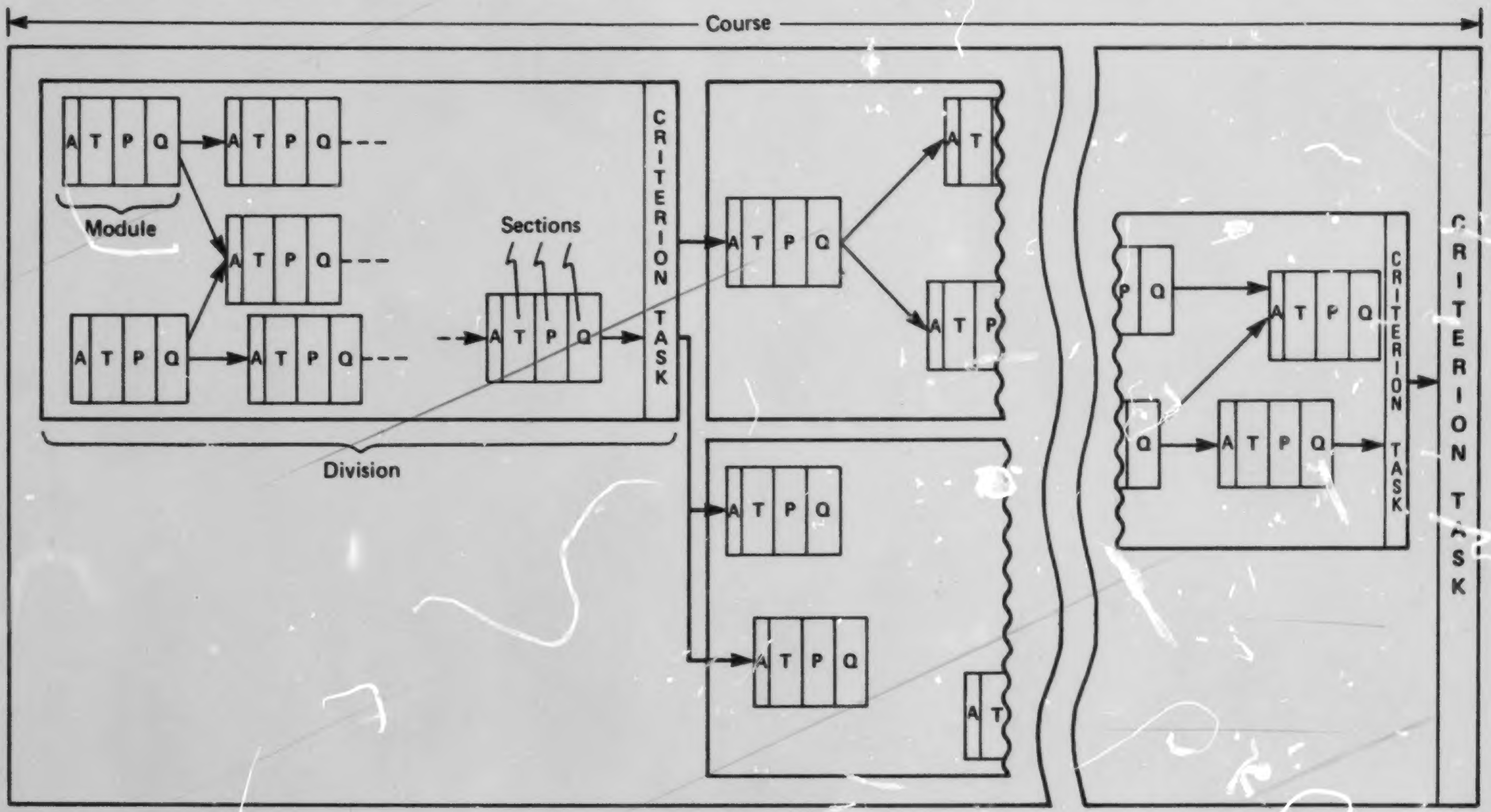
The hierarchical nature of the objectives demands that the course structure also be hierarchical. In order for the IDM to be implemented effectively and correctly, a clearly defined structure of interrelationships must exist so that the IDM (operationalized through the Interface) can select and retrieve stored instructional information for display to students. The ordering of the instruction is the prerogative of the author within the parameters of the IDM. The description of the requisite structural course schema which follows was adapted in part from instructional material developed by HumRRO for the Department of the Army.[1]

> The "course" is at the apex and contains all other elements of the schema. The term "course" connotes the major unit of subject-matter content. It is the complete instructional treatment (all the instructional influences) intended to produce in a specified student population a set of specified behavioral capabilities (terminal behavioral objectives).
>
> Because the set of terminal behavioral objectives for a course is likely to be quite large and complex in structure, it is necessary to group the objectives into smaller sets or subsets. The instructional treatments designed to achieve these various subsets of behavioral objectives constitute a "division." Figure 4 shows that divisions exist entirely *within* the course. It is also suggested that divisions within a course need not necessarily follow each other in a strict, linear sequence (although divisions in the COBOL2 course must follow sequentially); it is suggested that, for example, either the second or the third division of a course might legitimately follow the first division.
>
> The set of terminal behavioral objectives for a division is still likely to be sufficiently large and complex to require a further partitioning of behavioral objectives—from subsets into sub-subsets. This produces a "module." As with divisions, modules can sometimes be presented in various orders at different times. A list of modules in INTRO and COBOL2 is presented in Appendices B and C.
>
> Modules contain "sections," which almost always follow each other in the order depicted in Figure 4. The first one is the A(Administrative)-section, which contains a variety of administrative documentation that is not all visible to a student, but serves primarily to provide management information. The second one is the T(Telling)-section, which serves to present the relevant subject matter to the student. It is followed by the P(Practice)-Session, which permits the student to practice the relevant aspects of the behavior first described or explained in the T-section. In turn, the P-section is followed by the Q(Quiz)-section, which tests for the successful achievement of the specified behavioral objectives.
>
> A module can have several versions of each section. The limitation is, of course, that each version covers the same basic material and therefore teaches the same objective. The differences between versions are differences of form rather than content. For example: Version 1 of a T-section (T_1) may require

---

[1] Felix F. Kopstein and the Instructional Programming Staff, "COBOL Programming: Reader's Guide to the Instructional Content for Computer Administration," HumRRO research product, June 1972 (this was part of the total IMPACT courseware package sent to U.S. Continental Army Command).

Course Structure—Divisions, Modules, and Sections


Course
Module
Division
Sections
A T P Q
CRITERION TASK

A = Administrative Section

T = Telling Section

P = Practice Section

Q = Quiz Section

Adapted from "COBOL Programming: Reader's Guide to the Instructional Content for Computer Administration," Felix F. Kopstein and the Instructional Programming Staff of Project IMPACT, HumRRO research product, June 1972.

Figure 4

extensive reading, whereas Version 2 of the T-section (T_2) for the same module may be highly pictorial.

Two additional features should be noted in Figure 4. First, just as Q-sections verify the attainment of behavioral objectives for a module, a criterion test, or task, verifies the attainment of specified behavioral objectives for divisions and for the entire course. Second, in Figure 4, the jagged lines through divisions and the course are intended to suggest that some "middle" portion has been omitted so that only the beginning and the end of the course are depicted. Again, the structuring of divisions and modules (Figure 4) is illustrative only; it does not represent any specific course.

The first four levels of the schema are summarized as follows:

Level I — Course
Level II — Division
Level III — Module
Level IV — Section (A,T,P,Q).

While the schematic arrangement of instructional content being described here is related to and defined by Project IMPACT's IDM design, the levels that have been described so far have a broader applicability and validity with respect to the design of any instructional content. The levels to be defined and described next are far more heavily dependent upon the specific hardware/software subsystem (HSSS) upon which the course was implemented.

Within sections, instructional content is stored in "instructional units." An instructional unit is a grouping of closely related instructional presentations or components of a complete instructional communication or message. In practice, this might include one or more displays of text on the CRT and/or one or more frames of film projected via the Perceptoscope (P-scope).

The schematic representation is continued in Figure 5, and it is suggested that readers follow the explanation by referring to the schema in Figure [illegible].

The instructional communications or messages grouped within an instructional unit are called "pages" individually. Again, the internal structure within sections and instructional units illustrated in Figure 5, is arbitrary and intended only to suggest that many different arrangements are possible.

Pages are then composed of elements. There are two types of elements, master and slave. Master elements are templates which format the pages (Figure 6). They usually contain text, but, in addition, they provide preformatted areas (called blocks) on the CRT screen into which slave elements and/or answer spaces can be displayed. A page consists then of a master element and all of its associated slave elements.

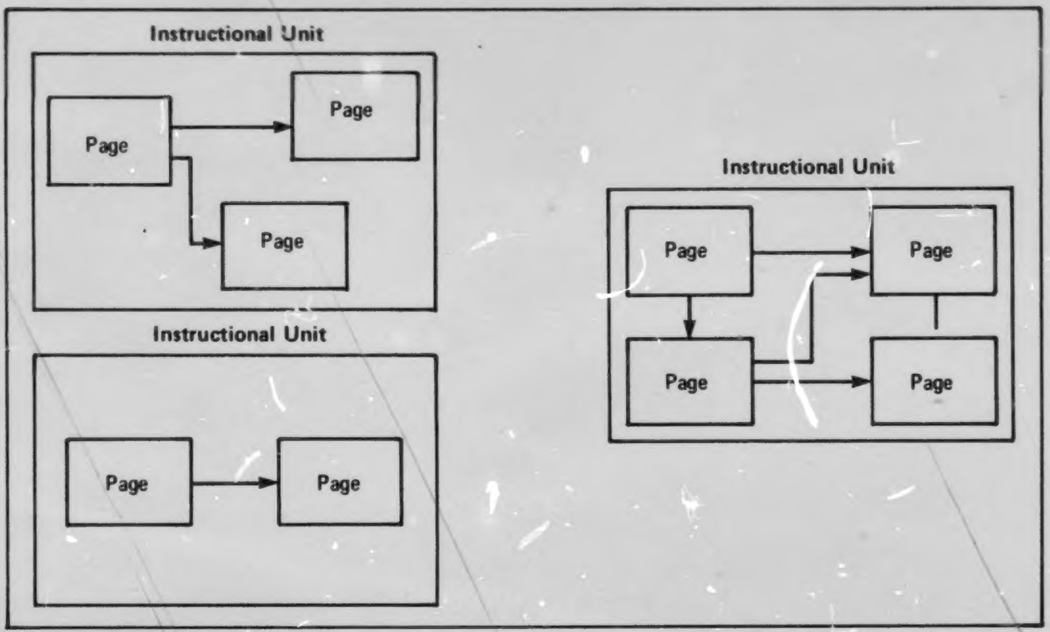
In general, the master and slave elements are stored on two separate files (D-file for master elements and F-file for slave elements) for ease of identification and manipulation. This does not mean that a D-file element cannot be a slave element or an F-file element a master.

The rules of application for the course structure just described are in the IDM. The author must define, for example, the modular prerequisite structure for the course in a manner interpretable by the IDM. The IDM is responsible for the making of decisions as to what module the student should see at the appropriate time.

INSTRUCTIONAL DECISION MODEL (IDM) FEATURES

By selectively specifying when the features of the IDM should be active or inactive, the author can regulate the amount of learner or system control in the learning environment. These features are expressed to the student as parts of the "INFO-" option.

Sample Page Groupings Within Instructional Units of a Section



Adapted from "COBOL Programming: Reader's Guide to the Instructional Content for Computer Administration," Felix F. Kopstein and the Instructional Programming Staff of Project IMPACT, HumRRO research product, June 1972.

Figure 5

"INFO-" refers to the five-character code that appears in the lower left corner of most master elements. These five characters must precede any IDM request made by the student. All features of the INFO- option are not available during a quiz. The requests and subsequent action taken through the functional IDM (Figure 7 flowcharts these interactions) are as follows:

(1) Any Word. INFO- followed by any word, except those in Items (2)-(6), results in either the definition of that word or a message saying that the word cannot be found being displayed to the student. This feature is called the glossary. The full list of words available on-line (I-file) is presented in Appendix G. The criteria for adding new words has generally been three requests made by three different students.

Diagnostic Code. In the COBOL2 course there are four modules in which the student has to correct his program. Error messages are labeled in COBOL by an alphanumeric code. When the student types the numeric part of the code after INFO- and transmits this request to the computer, the computer displays on the student's CRT a short explanation of the error message and suggestions as to correcting the error.

(2) @@. INFO- followed by @@ indicates that what follows is a student comment. Students can input information to the system (by commenting) at any time, except during a quiz. The student's comment is recorded for later inspection (by administrators, instructional designers, or authors) and the student immediately receives an acknowledgement of his comment and a note (on the CRT) that it has been recorded.

Sample Page Components

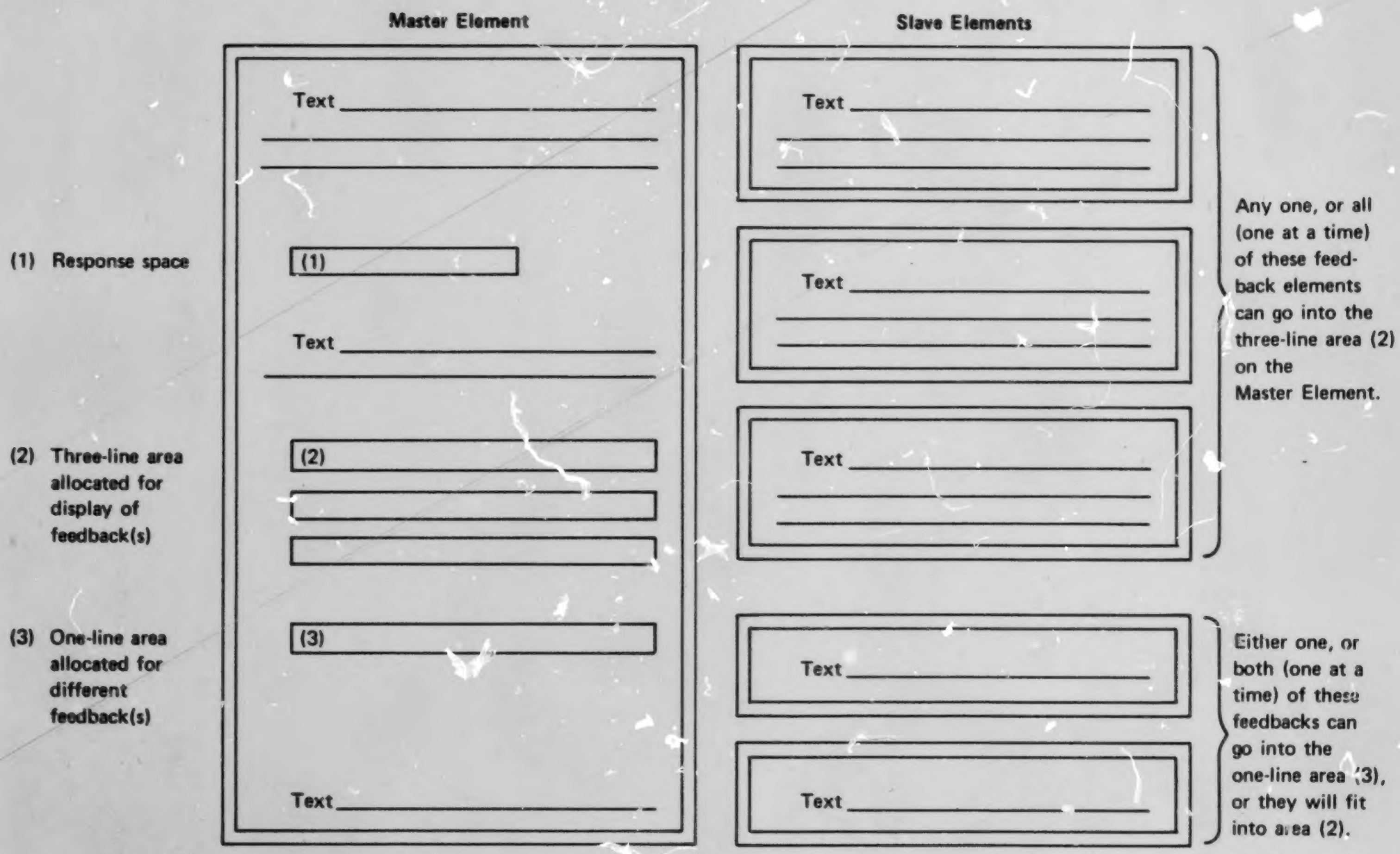

Master Element
Slave Elements
(1) Response space
(2) Three-line area allocated for display of feedback(s)
(3) One-line area allocated for different feedback(s)
Text
(1)
Text
(2)
(3)
Text
Text
Text
Text
Text
Text
Any one, or all (one at a time) of these feedback elements can go into the three-line area (2) on the Master Element.
Either one, or both (one at a time) of these feedbacks can go into the one-line area (3), or they will fit into area (2).

Figure 6

Flowchart of Instructional Decision Model (IDM) Features Under Learner Control

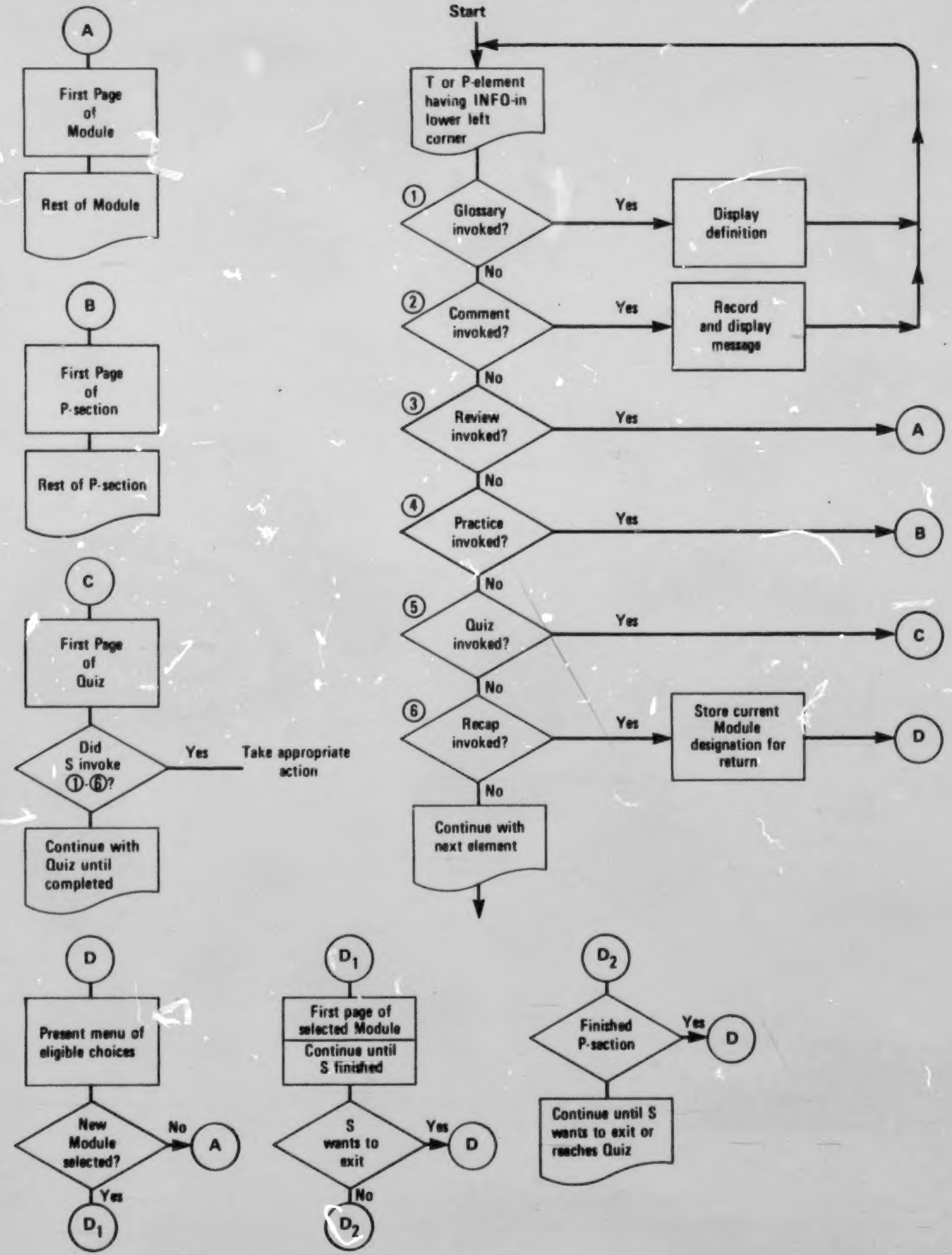

Start
T or P-element having INFO-in lower left corner
① Glossary invoked?
Yes
Display definition
No
② Comment invoked?
Yes
Record and display message
No
③ Review invoked?
Yes
A
No
④ Practice invoked?
Yes
B
No
⑤ Quiz invoked?
Yes
C
No
⑥ Recap invoked?
Yes
Store current Module designation for return
D
No
Continue with next element
A
First Page of Module
Rest of Module
B
First Page of P-section
Rest of P-section
C
First Page of Quiz
Did S invoke ①-⑥?
Yes
Take appropriate action
Continue with Quiz until completed
D
Present menu of eligible choices
New Module selected?
No
A
Yes
D_1
D_1
First page of selected Module
Continue until S finished
S wants to exit
Yes
D
No
D_2
D_2
Finished P-section
Yes
D
Continue until S wants to exit or reaches Quiz

Figure 7

3) REVIEW. INFO- followed by the word REVIEW causes the Interface to return the student to the first page of his current module and he can go through the module again.

(4) PRACTICE. INFO- followed by the word PRACTICE causes the Interface to take the student to the first page of the practice section of his current module. The student can request PRACTICE as often as he wants before the quiz section, so that it may be desirable for an author to write several alternative practice sections so that the student sees different exercises each time he goes through the practice section.

(5) QUIZ. INFO- followed by the word QUIZ causes the Interface to take the student from anywhere in the T- or P-section to the first page of the quiz. Thus, the optimistic and/or knowledgeable student can skip any instruction which he feels he does not need and can immediately demonstrate his capabilities for dealing with the criterion task/test. The first page in each quiz, however, gives the student one last chance to use any other feature of the INFO- option. If the student successfully completes the quiz, he will continue with the next module, but if he fails he is returned to the first page of his current module.

(6) RECAP. INFO- followed by RECAP causes the Interface to allow the student to recapitulate any division or module he has seen previously. This enables the student to review requisite material whenever he feels it is relevant to his current module. The student gets to the module he needs through a series of "menus" (lists of completed divisions and/or modules). He first chooses the division he wishes to return to and then gets a module menu from within that division. The student can exit from a recapped module any time he wishes and he goes back to the menu so that he can either recapitulate other modules or return to his current module.

(7) ROUTE. ROUTE is unlike the features listed above in that it is not associated with INFO- and the student can only apply it when the system allows. Notice in Figure 8 that a student can take either module F, G, or H after module E. He must, however, take all three modules before he reaches module I. So, at the end of module E the student will see a menu and he can choose the order in which he will study modules F, G, and H. This student choice of path through a division (course) is the essence of the ROUTE feature.

INSTRUCTIONAL TREATMENT

Because of the student inquiry and mobility techniques described above and the requisite course structure, an author must write material for use with this IDM in a very specific manner. The modules must be as independent as possible from other modules in the course. Thus, modules deal with either single or closely related objectives (see Appendix C). This encourages the student to use RECAP because he will find only material relevant to the information he needs when he requests a module. Also, since the instruction is written to stand alone (no transition between modules), it will make sense to the student when he comes back to see just that module.

The independence of modules is also important when the student employs the ROUTE feature of the course. Regardless of how the individual student traverses the modules, the final criterion task is the same. Therefore, no matter in what order he sees the modules, each module must be capable of independently giving the student enough information, in a readily assimilable manner, so that the student can achieve the module objective without necessarily having taken any other module (unless another module is specified by the author as a prerequisite).

The sections within a module must also be written to accommodate the features of the IDM. A T-section should be written so that the general concepts are introduced first

Choicepoint in Course ROUTE

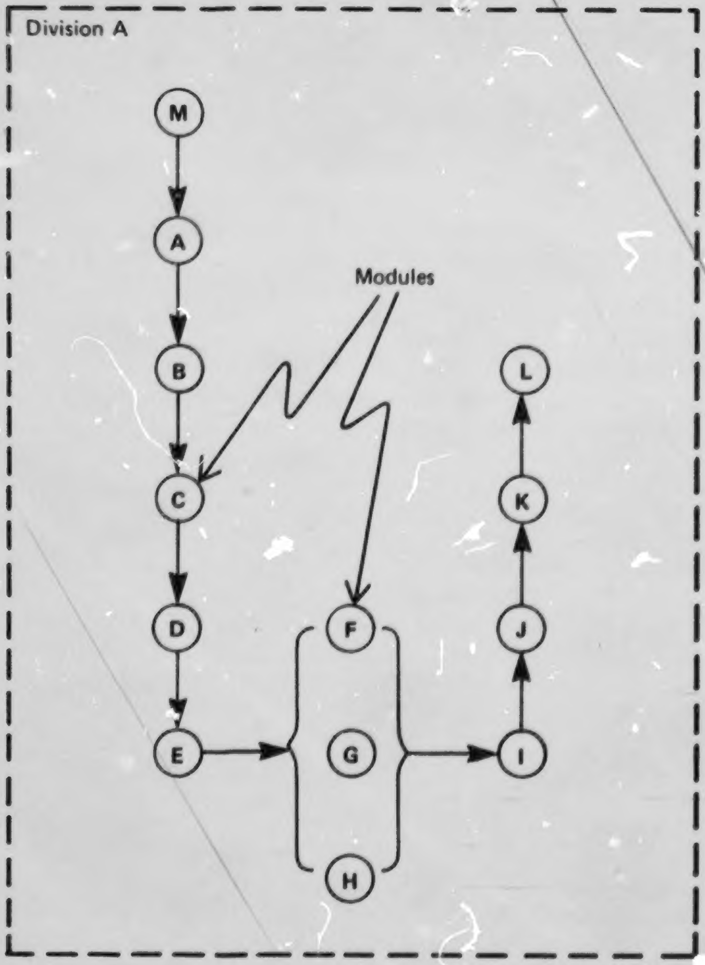


Figure 8

and, as the student reads further, the specifics are explained in detail. This technique enables the student to discern as early as possible whether he thinks he has sufficient mastery to skip to either the P- or Q-section. It also enables the student in REVIEW or RECAP to find the information he needs as early as possible and thus spend minimal time going back over material he has already seen.

The P-section provides content and format preparation for the quiz, in that the format of the quiz questions should be represented in at least one practice item and the practice items reflect the enabling objectives for the module criterion task. The P-section also provides drill and practice, and detailed explanation (in the form of feedbacks) to complement and expand on the information presented in the T-section.

Although some practice-type items may appear in a T-section, the bulk of the practical exercises should appear in a distinct P-section. These enable the student who invokes the PRACTICE feature to be taken to the bulk of the exercises in a module, regardless of whether he is coming from a T-section because he would rather "learn by doing" than by reading, or from a Q-section (first page) for some last-minute drill and practice before testing.

The structure of a P-section should resemble that of the T-section in that questions dealing with the general topic should be first, followed by questions on specific enabling objectives (for the module). This is, again, so that the student can be tested as quickly as he feels he is ready. The content of quiz questions should not be duplicated in the P-section; however, it is reasonable to have practice items which require the same mechanics of answering that are required in the Q-section. In this way the quiz can more accurately test the content of the objective and not be affected by the idiosyncracies of question presentation or methods of student response.

According to the IDM, the Q-section must also be a discrete, self-contained section. The content must include explicit directions to the student on how to answer the questions that test the module objective. Additionally, the author must specify the points that the student can earn, based on two scores. The first is a raw score that is unique in assignment to each quiz. (For more information on raw scores assignment, see page 35.) The author must then write specifications to transform raw scores into IDM scores. IDM scores are a standardized set of scores (1-10) used in every quiz. An IDM score of six is always the minimum passing score. It usually corresponds to a raw score equivalent to approximately 80% achievement on the quiz. The IDM is the score which the IDM references to make decisions regarding quiz feedback and module selection.

HARDWARE/SOFTWARE CONSIDERATIONS

The instructional text is also affected by hardware/software considerations. The primary affectors are the peripheral hardware devices used for the presentation of instruction to the student.

INPUT/OUTPUT TERMINAL

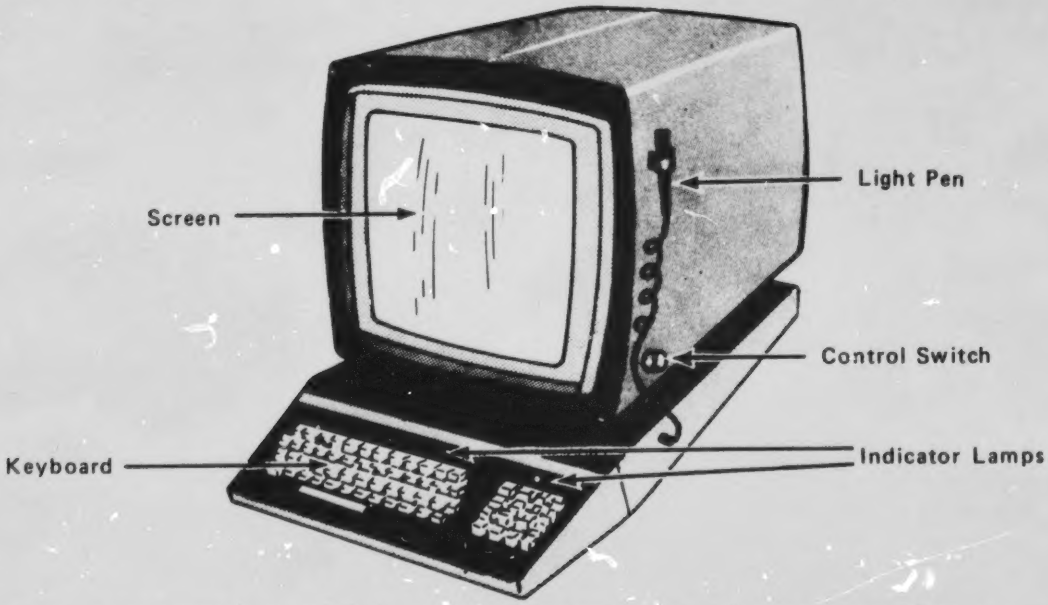
The cathode ray tube (CRT) display terminal is a device commonly used in computerized training for presenting instructional text to the student and receiving his overt responses to questions and other course materials. A CRT configuration generally encompasses not only a screen (like a television set), but also a keyboard (like a typewriter). Other features may be included, such as the light pen that is used directly on the picture screen to move or mark characters in the text.

The CRT employed for the IMPACT courses is a Sanders 720[1], used in a horizontal format (Figure 9). The CRT screen can hold up to 1024 characters (including format characters) in an 84-column by 32-row grid (2688 possible positions). The terminal has a light pen attached which serves to move the cursor (a blinking dash which is a place marker on the screen) to wherever the student wants it or is required to move it. For example: the system has positioned the cursor in readiness for the student to answer a question, but the student decides that he wants to REVIEW before answering. He would move the cursor, using the light pen, to the dash in INFO-, space once, type REVIEW, and transmit this request to the computer.

---

[1] Identification of products is for research documentation purposes only; this does not constitute an official endorsement by either HumRRO or the Department of the Army.

CRT Display Terminal

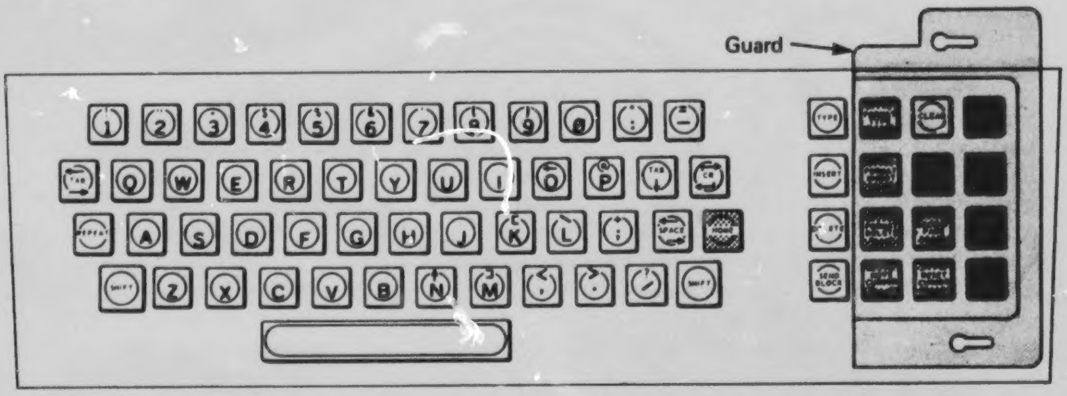


Source: Project IMPACT Staff, "COBOL Programming: Student Reference Manual," HumRRO research product, October 1972.

Figure 9

The keyboard, used by both student and author, is shown in Figure 10. The character set includes uppercase alpha characters, numerics (0-9) on separate keys, some special characters which appear above the alphanumeric keys, format characters, and mode (type, insert, delete) keys. Because the alpha characters are all uppercase, it is advisable for better legibility to skip a row between lines of text.

CRT Keyboard



Source: Project IMPACT Staff, "COBOL Programming: Student Reference Manual," HumRRO research product, October 1972.

Figure 10

Notice the guard over some of the mode keys in Figure 10. The keys that are covered by the guard are not available to students. They are used by authors to structure "pages" on the CRT. In author (format) mode, for example, when an author strikes the vertical tab key, a vertical tab character is generated, using one character from the pool of 1024 available for that element, and the cursor is immediately moved down four rows and placed in column one. All carriage keys generate screen structuring characters (called format characters) in author mode, but do not restructure the screen in student mode.

The author has a technique available to him known as blocking, allowing the author to break an element into logical units called blocks. Blocks can consist of text (which can be scattered throughout the screen) or empty spaces (for student answers or slave element display) or a combination of both. When a student answers a question, he transmits his answer to the computer by sending the answer block to the machine so its contents can be analyzed. Because of Coursewriter III buffer lengths, the answer block can be no longer than 100 characters, including format characters. The analysis process is initiated when the student strikes the SEND BLOCK key on his keyboard (Figure 10) and his instructions to do so, in the text, will read either PRESS SEND or PS. Simple page turning is also accomplished by the student pressing the SEND BLOCK key.

VISUAL SYSTEM

Because of the lack of an efficient graphics capability—simple graphics could be done using alphanumeric and special characters—on the CRT, an auxiliary visual device was included in the student terminal configuration. The device chosen was the Perceptoscope (P-scope) which projected images through a rear projection system from a 16mm film onto a screen which was located next to the CRT screen. The P-scope is a random-access machine which is entirely under computer control (author specified) through a Digital Device Controller (DDC). It is through the DDC that the film and CRT screen presentations are synchronized.

As used in Project IMPACT,[1] the P-scope is well adapted to image presentation but is limited insofar as text presentation is concerned. For example, there is a limitation of 24 lines of 14-point lettering on any P-scope visual involving text. If possible, the P-scope should not be used to present detailed instructions to the students, because any changes in the instruction, either because of an error or for the sake of clarification, are difficult, time-consuming, and costly.

An author considering choosing this medium for his visual presentation should note the following factors:

(1) If color is wanted, either for aesthetic reasons or for emphasis, the P-scope is the appropriate medium.

(2) The P-scope has broad illustrative and graphic capability. Except for the text constraint, the only limitation on the P-scope visual is the creativity of the artist.

(3) If an animation or movie sequence is needed within the subject matter being taught, the P-scope provides the capability.

(4) If the CRT is already in use for a given display, the P-scope may have to be used for text presentation.

(5) A dynamic or changing presentation other than the CRT might be useful. (There are several instances in our courses where the P-scope has become the dynamic means of presentation and the CRT image holds the static image.)

(6) The P-scope provides another controlled means (besides the CRT) of presenting the information to the student.

---

[1] The P-scope film used for the INTRO and COBOL2 courses is available in 16mm and 35mm.

AUXILIARY COURSE MATERIALS

In general, the term "auxiliary" may be applied to any instructional materials that supplement the main body of text. However, in a computerized learning system, the term is perhaps more aptly applied to off-line course materials, meaning supplements not physically connected to the computer or any other devices in the total configuration.

In Project IMPACT, on-line supplements to the main body of text included the visuals presented to the student with the Perceptoscope System and the glossary available on the CRT through the INFO- feature. Off-line supplements included the directions for keypunching and other reference material in the Student Ready Room, as well as the Student Reference Manual.

Visuals were also available as off-line supplements in the form of filmstrips to be used with manually operated projectors in case of system breakdowns. Bound hard-copy printouts of the glossary also served as back-up and were used like a dictionary when system problems so indicated.

The functional context of the COBOL programmer's job normally requires considerable facility in the use of reference manuals. For this reason, and because it also serves as a tertiary display of instructional information, the Student Reference Manual was incorporated as the primary off-line supplement to Project IMPACT and an integral part of INTRO and COBOL2 courses.

Since the student has the manual with him almost all the time, the author must be careful that he does not negate quiz results, either in a current module or a future one, by putting too much course-specific information in the manual. If the author of a quiz knows that the answers to the quiz can be found in the manual, he can decide that the proctor should remove the manual while the student is taking the quiz. Therefore, the author should make certain that there is no cueing from any part of the manual that would invalidate criterion test results.

Supplements may also serve as essential reference tools for instructional and software programmers during the development and implementation of all other course materials. Therefore, the need is obvious for planning, drafting, and distributing the contents of such supplements very early in the process of instructional design. Equally important is controlling and maintaining all existing copies, in order to ensure accurate and complete updates of any changes. Some kind of cross-reference system should also be established and maintained to facilitate checking and updating directions to the student for using the supplements.

STAFF CONSIDERATIONS

The Project IMPACT approach to the development of instructional materials has been to use a multidisciplinary team. The team capabilities should include instructional design, subject-matter expertise, instructional programming, editorial skills, hardware and software expertise, media selection and utilization, and production and management skills. Project IMPACT experience has shown that a different person is not needed for each of the skills, but the person who is "wearing more than one hat" must be competent in all assigned functions.

The instructional designer establishes behavioral requirements, encodes them into instructional objectives, and designs the IDM that will best serve the needs of the students and the subject matter. The subject-matter expert provides "raw" content to the instructional designer and instructional programmer and serves as a consultant when the programmer structures the content for implementation in a CAI mode. The instructional programmer is a specialist in using computers effectively and efficiently for instruction.

The editor reviews the draft instruction, which is usually delivered on a paper representation of the CRT screen. The manuscript format used by Project IMPACT is shown in Sample 1 (Appendix E). When it is filled out, it contains all the information for a particular element—text, characteristics (to be entered in the title listings), and on the right side the production flow. The editor is examining the material for accuracy, consistency (inter- *and* intra-modular) and style.

Unless the style of a piece of instruction is especially aberrant, the author is free to employ and/or create any intra-modular techniques he chooses (within the constraints of the IDM). The material next goes to the technicians who input the instructional text and program the instructional logic Someone also has to assume the responsibility of a production manager to assure the timely passage of the instructional package through the production chain.

Because of the volume of material produced by Project IMPACT, most of the above functions were allocated to different people. The amount of manpower required will vary according to the amount of materials produced, but all the skills mentioned above must be present to effectively manage instructional materials through the production process.

Another element which affects the instructional text is the use of course administrators (proctors) during the actual operation of the course. Although their presence is needed primarily to solve logistical and mechanical problems for the students, they are required to have a knowledge of both the subject matter and the system. Therefore, they can be used to provide subject-matter assistance when an author desires it, and to check student assignments done off-line. To use the proctor during an instructional sequence, the author must define the conditions under which a student sees a message on the CRT to call the proctor. For example, some authors specified that the student should call the proctor for assistance when the student had incorrectly answered a practice item on two consecutive attempts. Other rules for generating proctor calls can, of course, be defined by authors. All planned proctor calls for a module are documented in the A-section of the module.

Chapter 3

INSTRUCTIONAL DOCUMENTATION

The term "documentation" means different things to different people. Depending on the point of view, it may cover products or programs, as well as records or reports prepared before, during, or after an activity. In a total system, the meaning is complicated by outputs from one activity serving as inputs to another.

In the discussion that follows, the term "documentation" refers to all records and reports that *describe* or *prescribe* the instructional materials developed in Project IMPACT. In each case, the resulting document has been made available in hard-copy form, even though it may have been composed via the computer system with some kind of man-machine interaction.

SYSTEM REQUIREMENTS

The need for accurate, detailed, and standardized record keeping became evident in the very early stages of course development in Project IMPACT. Every separate piece of instructional material had to be labeled. Each piece of a set had to be so identified. The sequence of each piece and set had to be stated. Alternative paths for anticipated and unanticipated student responses had to be formulated.

Programmed instruction and computer programming provided useful parameters in the beginning. Authors and input technicians also relied heavily on such well-known managerial tools as charts, checklists, procedure manuals, tables, and staff memoranda. Forms for every step in the process seemed to be the only way to keep track of the ever-increasing detail and handwork.

The problem of managing the documentation eventually became as critical as managing the instruction. There never seemed to be a complete and up-to-date set of design, development, and implementation documents at hand for quick and easy reference, and everybody needed them—authors, software programmers, proctors, research scientists, and work supervisors.

Solving this paperwork problem was one of the major accomplishments of the revision process of COBOL2. Standardized on-line documentation, built into a book-like format for each instructional module, made it unnecessary to refer to any other source of information. Once input into the computer system, the documentation could be managed like any other part of the CRT instructional material.

RESPONSE ANALYSIS DOCUMENTATION

The need to manage complex and voluminous course material with standardized documentation became especially apparent when an answer analysis for practice items with multiple feedbacks was written. Sample 2 (Appendix E) shows the on-line documentation for a hypothetical practice item. This documentation would not be seen by the student, but would exist in the instructional text files and be reproduced when the hardcopy form of these files is generated. Because the documentation was both repetitive and exhaustive, a system of mnemonics was used, as shown in Sample 2. The mnemonics are defined as follows (not all are shown in Sample 2).

CA[1] = The correct answer is. . .
WA = An anticipated wrong answer is. . .
WN = The nth anticipated wrong answer is. . .

CAF = Correct answer feedback. A CAF is provided when the student's answer is accepted as being correct. A CAF need not be given in every instance when the student has responded correctly, but it serves to emphasize the correctness of his response on appropriate occasions, or to stress, through repetition, the salient aspect of the subject-matter information. If no special comment is made when the student responds correctly, he simply advances to the next question or page, in which case the documentation could read "go to next A" or "go to nnA," where nn equals a page number and A indicates a master element.

TAF = Try again feedback. Often, after incorrectly answering a practice question on the first attempt, the student is allowed to reattempt the question. The TAF incorporates the "try again" message and usually some additional clues to aid the student in arriving at the correct answer.

TAN = A TAF written to correspond to a particular wrong answer (N) and/or a catch-all feedback for an unanticipated answer. The former type usually contains some information specifically explaining why the student's answer was wrong. The catch-all TAF usually states a general rule whereby the student can more readily arrive at the correct answer.

WAF = Wrong answer feedback. A WAF is usually provided when, after more than one response attempt, the student still has not succeeded in giving an entirely correct answer. Normally, the WAF will contain additional comment to point out to the student what the rule was for deriving the correct answer and/or eliminating some of the wrong answer possibilities. Often the WAF will contain the correct answer.

WAN = A WAF that is associated with a particular anticipated wrong answer (N). If the student gives this answer he is usually told, in the feedback, why his choice was wrong.

DAF = Diagnostic answer feedback. In the COBOL2 course, the student was required to make many COBOL coding responses. Feedback for this type of response often consisted of a TAF and a list of appropriate DAFs, which are "canned" feedbacks that relate to specific COBOL errors and are worded similarly to COBOL error (diagnostic) messages. (See Appendix F for the list of DAFs used in the COBOL2 course.)

ZAP = This is not a mnemonic. The ZAP feedbacks are canned feedbacks similar in size and structure (short, non-question specific text) to the DAFs, but not specifically related to the subject matter (see Appendix F).

ROT = Roll on text. This type of feedback can either be displayed to the student from the feedback file or as a literal embedded in the instructional logic. Sample 2 (Appendix E) tells the instructional coder to display or "roll on" the correct answer into the fifth logical block of the master element when the WAF is displayed.

These mnemonics appear throughout the CRT course materials on pages of Response Analysis Documentation (RADs if stored on the D-file and RAFs if stored on the F-file). They also form part of the title given to each piece of instruction input into the computer system.

---

[1] Although many of the mnemonics appear to be the same as Coursewriter (CW) OP codes, the resemblance is incidental, and no conclusion regarding program execution should be drawn.

TITLES AND THEIR USES

In Project IMPACT, titles were designed and developed by instructional and system programmers as a means of identifying and describing each piece of course material. Neither the disc storage location nor the Coursewriter-Instructional Programming (CWIP) label provided enough detail to describe the CRT text. Similar needs surfaced for visuals, terms, and diagnostics for the glossary file, and even pages in the Student Reference Manual.

Titles were standardized for each type of course material, but it was not necessary to have the same format for each piece of instruction. The design of titles for all CRT material (text and glossary) was determined in large part by the hardware/software subsystem. The concept was adapted as needed for visuals and the manual.

Each title evolved from directions supplied by the author to a coding form, and then to a punched card. During the developmental stage of the course, simple listings were made of these card records to keep everyone informed. As the course approached finalization, however, CRT titles were placed on-line with an Auto-Insert program and other programs which were written to output the entire contents of title cards in a more readable tabular form.

Titles and corresponding documentation have been used by instructional programmers of both courseware and software as documentation of production, as substitutes for flowcharting, and as indices for analyzing and checking content, logic, and storage locations.

Following are descriptions of the titles and title records for each type of course material developed in Project IMPACT.

CRT TEXT

The output record for listing titles and title documentation for CRT text contains 10 data fields (see Sample 3, Appendix E). The input records also contain additional administrative and system information. Only the contents of data fields 2 through 8 (columns 15 through 46) comprise the title per se and therefore appear on the hard-copy printouts of CRT textual elements.

The contents of the data fields in both punched cards and listings are explained below:

(1) Disc Location (columns 1-6). A six-character code which identifies the location on a magnetic disc where each element is stored. Both files (D and F) use the same code, such as D09999 or F09999 where 9999 is the four-digit disc location and the D or F designates the file in which the element is located.

(2) Display Label (columns 15-20). An alphanumeric code of six characters, the label or page number in the course. The first character represents the division; the second, the module; the third, the version (A-Z for the A-, T-, and P-sections and 1-9 for the Q-section); the fourth and fifth, a sequence or page number (starting with 00 for the first page of each module); the last, the sequence of elements within the page (A-E are generally reserved for master elements and the rest of the alphabet is used for slave elements).

Exceptions to label conventions occur in a group of elements known as the CAT-ZAP displays which are master and slave elements used extensively in the course. The letters CAT or CAP in the first three positions of the label signify catalog element or catalog performance summary, respectively. The remaining three positions are either the same as a normal label (i.e., a page number followed by a sequencing letter) or contain a

3-character mnemonic that describes the element. ZAP labels start with "Z" followed by the 5-character disc location of the element.

The following are examples of CAT-ZAP labels:

CAT01A - Catalog element.

CAPBA1 - Catalog performance summary for Division B, Module A, Objective 1.

CATCOM - Catalog element for the quiz computation sheet.

ZF1788 - ZAP element in disc location F1788.

(3) Coursewriter Status (column 22). A one-character numeric entry in the field which indicates the coursewriter status of the element. The most important information is whether the student sees the element or it exists for documentation (coded = student sees; uncoded = documentation). The explanation of each number follows:

0 - uncoded
1 - coded
2 - coded, but experimental
3 - coded, and contains one or more "blanking characters"
4 - coded, and not to be modified without consulting the editor or inputter, due to the peculiar nature of the display
5 - coded, and contains "blink" characters

(4) Section (column 24). A simple one-character designation indicating whether the element is from the A-, T-, P-, or Q-section.

(5) Purpose (columns 26-28). A three-character mnemonic as defined earlier. (See Appendix F for a complete list.)

(6) Number of Slave Elements (column 30). Information on number of slave elements associated with a given master element, including the uncoded RAF. The digits 0-9 are used to indicate the number of elements; if the number exceeds 9, the letter "X" is used.

(7) P-Scope Status (column 32). Status indicated by a 0, P, or S where:

0 = the P-scope is off.
P = a different P-scope comes on when this element appears.
S = the same P-scope stays on for this element.

(8) GO TO (columns 34-46). Indication as to which element does or could come next. A "+" appearing at end of the "GO TO" label indicates that the branching is complicated and the response analysis should be checked for a clearer understanding of the possible paths the student will take. Uncoded elements and CAT-ZAP elements have no information in the "GO TO" column.

(9) Blocking (columns 51-55). As explained earlier, blocks are the physical fragmentation of an element into fields. The blocks are numbered in the order in which they are created and for the most part an author can specify the number and location of the blocks in an element which he has written. There are, however, some standardized blocks which occur in most elements, and if they occur their order is predetermined (Sample 4, Appendix E). The blocking structure for any element is described by a 5-digit string and each position in this string has a specific meaning. If a zero appears in any given position in the string, that particular characteristic is missing in that particular element. The meaning of each position is as follows.

Position 1 - signifies the presence of an "INFO-" block, whether normal, off-set (one space character ahead of "INFO-"), or dummy ("INFO-" is not written in the block).

Position 2 - signifies the presence of a glossary block consisting of three full lines of space characters.

Position 3 - signifies the presence of a text block in the element. Slave elements have a zero in this position because there is no special block for text.

Position 4 - signifies the first block created for displaying slave elements on a master element. Usually, this block consists of three lines of space characters, double-spaced. For slave elements, this position specifies in which block the element will appear. For uncoded elements, such as RADs, this position has no meaning. An "X" may be used in the CAT-ZAP elements in this position to show that this element may appear in any block.

(10) Student Response (columns 57-59). A three-character mnemonic representing the type of response expected from the student. (A list of mnemonics is in Appendix F.)

Additional information on the COBOL2 title cards includes the following data fields that do not appear in Sample 3:

(11) Copied from. This 5-character string starts in column 66 and shows that the element is copied from a given disc location without change or with only minor changes. For example, IMB18G in disc location F02222 was copied directly from F1528 (the leading 0 is omitted).

(12) Repeated use. If column 75 contains the letter "R", the element is used more than once in the course in the same form.

(13) Auto-Insert. An "R" in column 80 indicates that, when the element was created, the full title (everything on the listing except the disc location, blocking, and student response) was not input and, therefore, the remainder has to be filled in via an Auto-Insert program.

System requirements are met in other columns on the title cards. Column 7 is blank, columns 8 through 13 contain the direction TITLE-, and single quotes in columns 14 and 46 enclose the actual title.

VISUALS

The instructional text for the P-scope is created, accessed, and modified differently from the text in the course files on disc. Therefore, other information is needed on the P-scope title records (Sample 5, Appendix E).

Figure 11 shows an excerpt from a full title listing for text and visuals in which P-scope title records are set off by indenting and double-spacing.

Data fields in the P-scope title records contain the following information:

(1) Cue Mark (columns 1-8). An alphanumeric code of 4 or 8 characters is used to indicate when the correct visual frame is ready to be projected. The code begins with "P" and ends with a 3-digit number for a single frame. For animation sequences, the code of the first frame in the sequence is followed by a slash and another 3-digit number that represents the last frame in the sequence. Images are referenced by cue marks that appear on the sound track, one per projected image and one randomly accessible (Figure 12). They are counted as the number away from an initial reference point.

(2) P-Scope Label (columns 11-16). Visual labels serve the same purpose as other course labels and follow the same format except that the last character in the string is always a "P". However, these labels do not always indicate in which module they are used. For example, a P-scope label beginning with "AJ" may be used elsewhere in the course and not just in Division A, Module J.

Excerpt from a Full Title Listing for Text and Visuals

```
DISC NO.  LABEL    CW ?  SECTION  PURPOSE  FOBKS  P-SCOPE  GO TO     BLOCKING  RESPONSE

D03277    AMA00A   1      A        HAT      0       P      AMA01A     12341     PSR
   P053     AMA00P        ON AMA00A OFF AMA01A     TITLE: COBOL PROGRAMMING
D03278    AMA01A   1      A        PRE      0       P      AMA04A     12341     PSR
   P054     AMA01P        ON AMA01A OFF AMA04A     TITLE: ABOUT THIS COURSE
D03279    AMA02A   0      A        DES      0       0                 12301     PSR
D03280    AMA03A   0      A        OBJ      0       0                 12301     PSR
D03281    AMA04A   1      A        DIR      0       0      AMA10A     12341     PSR
D03282    AMA05A   0      A        PRG      0       0                 12301     PSR
D00531    AMA05B   0      A        PRG      0       0                 12301     PSR
D03283    AMA06A   0      A        MAT      0       0                 12301     PSR
D03284    AMA08A   0      A        KFY      0       0                 12301     PSR
D03285    AMA10A   1      T        TRN      0       0      AMA12A     12301     PSR
D03286    AMA12A   1      T        EXP      0       0      AMA14A     12301     PSR
D03287    AMA14A   1      T        EXP      0       0      AMA16A     12301     PSR
D03288    AMA16A   1      T        EXP      0       0      AMA18A     12301     PSR
D03289    AMA18A   1      T        BRN      2       0      SPECIAL    12398     LPR
F01718    AMA18F   0      T        RAF      0       0                 00091     PSR
F01719    AMA18G   1      T        FAD      0       0      AMA20A     00091     PSR
D03290    AMA20A   1      T        EXP      0       0      AMA22A     12301     PSR
D03291    AMA22A   1      T        EXP      0       0      AMA24A     12301     PSR
D03292    AMA24A   1      T        EXP      0       0      AMA26A     12301     PSR
D03293    AMA26A   1      T        REC      3       0      AMA26G-H   12354     FC2
F01720    AMA26F   0      T        RAF      0       0                 00051     PSR
F01721    AMA26G   1      T        CAF      0       0      AMA28A     00051     PSR
F01722    AMA26H   1      T        WAF      0       0      AMA28A     00051     PSR
D03294    AMA28A   1      T        REC      2       0      AMA28G     12354     FC2
```

Figure 11

P-Scope Film

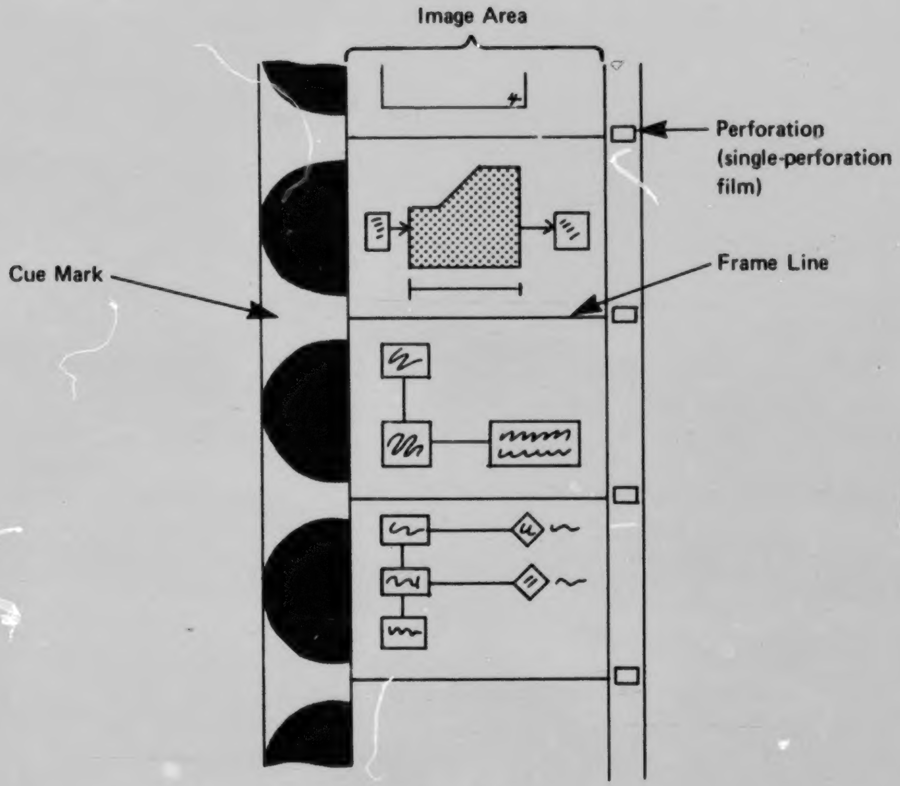


Figure 12

(3) On-Off (columns for On, 21-26; columns for Off, 32-37). This states the range of elements for which a given P-scope frame or sequence is shown to the student. The format is:

ON ABA00A OFF ABA04A

In this example, the visual frame comes on at display "ABA00A" and stays on until display "ABA04A" is encountered. In the title listing, P-scope records occur immediately after the first COBOL2 title for the display where the frame appears. Additionally, this COBOL2 title record will have a "P" in the P-scope column.

(4) Description (columns 41-79). Information in this area contains either the title of the frame or a brief description of the content of the frame.

GLOSSARY TERMS AND DIAGNOSTICS

From the beginning of instructional programming on-line in Project IMPACT, title cards were used as planning and research tools for the design and development of a glossary of terms and subsequent error messages.

The format for columns 1 through 47 had to conform to the same system requirements for CRT text titles. The letter Q was used in column 1 to designate the file in which glossary entries were stored. The 32 characters of the actual title field (columns 15 through 46) were used as the term key to the text which the student might request and retrieve via the INFO- option. Since only four of these characters were necessary for the error message code, the end-quote was placed (in column 19) so that the rest of the title field could be used to describe the type of problem that generated the diagnostic. The student did not see this documentation, but it does appear in a listing of the contents of the title field (see Appendix G).

The remaining columns of the title card were used for noting the kinds of information that are helpful, if not essential, to the orderly creation and maintenance of any lexicon. Examples of data recorded include the source of the term, the place of its introduction or request in the course, and cross-references to other glossary entries or to other course materials.

Sources of the glossary included the basic vocabulary of the subject matter (COBOL reserved words and related terms), other key terms used in the instructional materials, and error messages generated by student programs. Title card documentation of student requests made it possible to add or delete terms on the basis of actual need.

STUDENT REFERENCE MANUAL

Modified versions of the title card were used during several stages of course development as working documentation for the production of the manual and as indices to manual references in the other instructional materials. If an index to the manual itself were needed, title cards for the glossary could easily be adapted for this purpose.

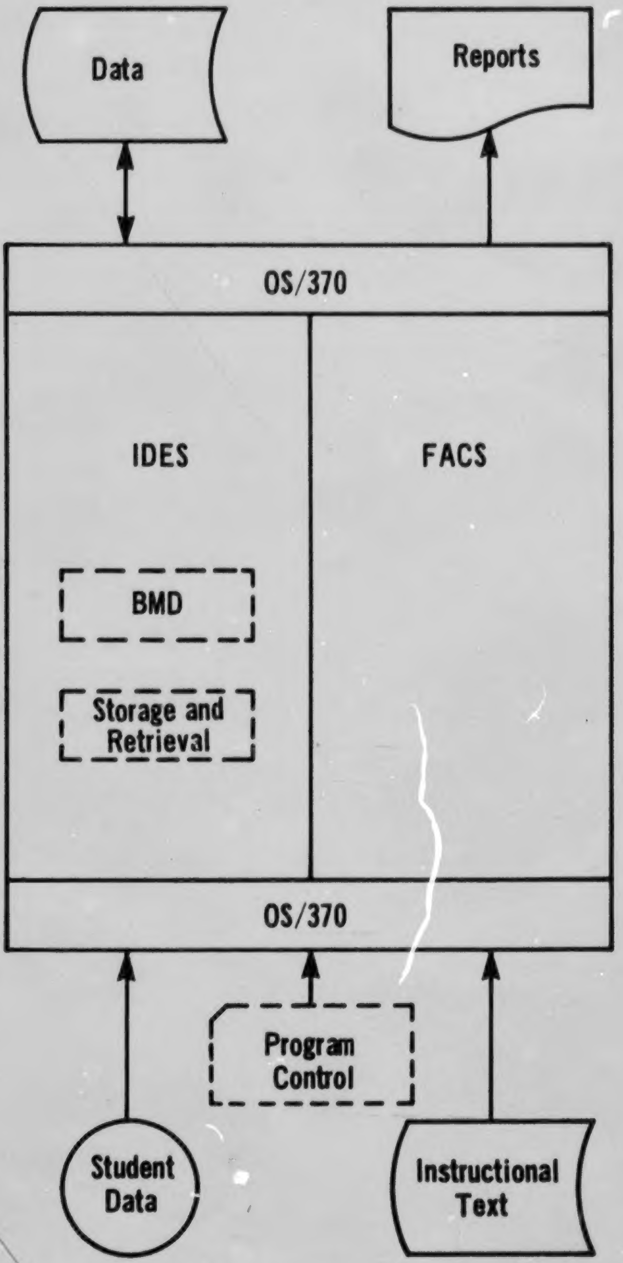
FACS AND MINI-FACS

The bulk of course-specific documentation is embedded in the instructional text itself and stored on-line. The student does not see this documentation, but it can be accessed by course authors for modification and/or viewing in the same manner as any other element. Likewise, these elements appear on course title listings and can be easily identified through the mnemonic field and the CW? field (0 in the CW? field, see Sample 3).

Most of these items can be found in one of three specific places: in the A-section, after questions, and in the beginning of the Q-section. It is not always desirable, however, to work with and inspect either the instructional or documentation items on-line. Therefore, a hard-copy, off-line repreoduction of the instructional text files was needed and this was accomplished via the File Activity Control System (FACS). The instructional text files references in Figure 13 are the same instructional text files shown in Figure 2.

FACS and MINI-FACS are discussed further in Chapter 4.

Off-Line Software Components



Data
Reports
OS/370
IDES
FACS
BMD
Storage and Retrieval
OS/370
Program Control
Student Data
Instructional Text

Source: John Selzer and Jean Garneau, *Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction Subsystem*, HumRRO Technical Report 72-21, August 1972.

Figure 13

Chapter 4

INSTRUCTIONAL MATERIALS

The set of instructional materials for computerized training in Project IMPACT consisted of the following products:

Text
Visuals
Glossary
Student Reference Manual
Supplies

The complete training package included products from the other subsystems of the project. As mentioned earlier, however, the scope of this document is limited to the actual courseware.

Each product in the courseware package will be discussed in detail.

TEXT

The hard-copy output of the File Activity Control System (FACS) is an accurate representation of how the CRT image of that element looks to the student (exception: middle dots appear as crosshatching on FACS output, Sample 6, Appendix E). Additionally, by including other kinds of documentation on the FACS output, an author can inspect that part of the element structure which is not seen by the student. For example: Under the column titled "number characters" in Sample 6, one finds three rows which read

84
84
83

There are no visible characters to the left of this designation (specifically within the rectangle enclosed by dashes), so the author can legitimately surmise that this is an area containing spaces that are probably reserved for the display of feedback to the student. This supposition is verified by further inspection of other areas on the FACS output as explained in the following material, excerpted in part from the instructional material in the IMPACT courseware package.[1]

> In the top margin, the first entry (top left corner) refers to the date on which that page of FACS documentation was printed. Next to it appear two entries, FACS and FORMATS, which relate to the software that generates FACS. Similarly, the entries following "OPTIONS:" constitute parameters of the program and specify which file (D, F, Q), whether the Page in question is formatted, whether the entire content of the file is to be printed, and whether a title is to be printed out. The number appearing in the upper-right corner is the page number of the FACS run and *does not represent any sequencing of the instructional content.*
>
> The second line from the top contains headings for variable entries in the third line. The DISPLAY NUMBER gives the file and disc location in which the text is stored. The

---

[1] Felix F. Kopstein and the Instructional Programming Staff, "COBOL Programming: Reader's Guide to the Instructional Content for Computer Administration," HumRRO research product, June 1972.

> TITLE has been explained as being a combination of other entries. The OPERATION is the latest one for the segment of text in question and specifies the nature of that operation (e.g., created, modified, copied). DATE and TIME tell exactly when this last operation occurred.
>
> The entry for AUTHOR refers to the person who inputs this text for magnetic storage on discs (not to the person who generated the text).
>
> A count of the horizontal dashed lines forming the top and bottom of the "box" shows that there are 84, with every tenth one numerically coded. This corresponds to the maximum length of 84 Characters within a Row that may be displayed on the CRT. Along the left side of the box 32 Rows have been labeled, corresponding to the maximum number of rows (independent of the number of Characters within any one Row) that may be displayed on the CRT.

The text inside these dashed lines is the actual formatted CRT text as seen by the student. The page label (BCA84 in Sample 6) is invisible to the student as it is surrounded by control characters which operate from the DDC set of commands. Very often slave elements will appear to be misformatted in FACS output (Sample 7, Appendix E), but they are constructed so that they fit into pre-formatted blocks on the master element and are thereby readable (in this case a two-row feedback block of 67 characters per row).

> Next to the box in the right margin of the FACS format are two columns headed "number of characters" and "number of control characters" respectively. The first of these columns provides a count for the number of characters each Row contains. The total of all Characters appearing in all Rows is also given. Similarly, in the next column, the number of control Characters is given. Control Characters are invisible on the CRT screen. Their function is to control the format (e.g., indention of a Line, location of a Block) of the text appearing on the CRT. A summary of the number of Blocks used or reserved within a particular Page is given at the left below the Row numeration. The number shown is always one less than the true number.
>
> The bottom margin is essentially given over to recording the true contents—not merely the visible ones—of the delay line that refreshes the visible display on the CRT. The true sequence of Characters on that delay line is shown, including the invisible control Characters (underlines).

An *M* indicates the beginning of a block and the blocks appear in the delay line in numerical order, *not* in their apparent screen order (e.g., INFO- always appears on the bottom of the CRT screen, but always is in Block 1). As is evident in Sample 6, Block 2, although invisible within the rectangle, shows up as three lines of blanks. Inspection of the delay line is also helpful to the author who wants to verify that the answer blocks are within the 100-character limit imposed by CW (answer blocks in excess of 100 characters cause incorrect answer analysis and scrambled screens).

The remaining information on the FACS output is a duplication of the information that appears in the top margin. Its second location in the lower part of the page is to facilitate "thumbing-through" the volumes of FACS output. FACS volumes are arranged in ascending order by Division, Module, Section, and Page. (AAA00A precedes ABA00A precedes ABB02A precedes BAA00A precedes BAA202A, etc.). The order of the FACS volumes, of course, is not necessarily the order in which the student views the material since the latter is a function of IDM and student selection and control.

An alternative form of FACS output is available, called MINI-FACS. MINI-FACS output is shown in Sample 8 (Appendix E). All it contains is the text of an element, the course name with which the element is associated, the date, and the full title of the element. This format is useful for people interested solely in the content of a course rather than operationalizing a course.

The format of the FACS (or MINI-FACS) pages is constant regardless of whether A-, T-, P-, or Q-section pages are being output. The content of the sections is obviously different because of the differences in the intent of each of the sections, the IDM design requirements, and documentation requirements for course administration.

The A-section of a module contains the general documentation for the module. There are 11 standardized pages in an A-section. Although most of these pages are included for course administrators, an author can choose which page he wants the students to see. Appendix E contains samples of the 11 standardized pages described below:

Page 00 — This is the title page for a module. It contains the name and letter designation for the module. (Sample 9)

Page 01 — This is the prerequisite page for the module. It lists those modules which the student must have passed to reach that point. (Sample 10)

Page 02 — This page contains a brief description of the module and mentions any special features unique to the module. (Sample 11)

Page 03 — This page documents the objectives for the module, in behavioral terms. (Sample 12. There is also a full list of COBOL2 objectives in Appendix C)

Page 04 — This page provides directions to the student—both those which are unique to the module and reminders as to the features of the INFO-option that are available. Page 04 also lists the materials which the student will need to successfully complete the module. (Sample 13, Sample 14)

Page 05 — This page documents all anticipated proctor-student interactions within the module. It lists the location in the module when the "proctor-call" will occur and the appropriate proctor response to the student's request. (Sample 15)

Page 06 — This is the auxiliary materials page for the module. It indicates whether the student manual is referenced, and which P-scope visuals are associated with which on-line text components (duplication of "on-off" fields from P-scope title records). (Sample 16)

Page 07 — This page is used only in occasional modules in the COBOL2 course. When most of the questions in a module require the same type of response analysis, this page is used to document the analyses. (Sample 17-18)

Page 08 — This page lists all the key terms—reflecting concepts which are to be introduced or emphasized in the module. (Sample 19)

Page 09A — This page lists all the special terms (programmer-supplied-names in the COBOL2 course) which are used in the T- and P-sections for the module. (Sample 20)

Page 09B — Same as 09A, except the special terms are used in the Q-section only. (Sample 21)

Page 10A — This is the transition page to the beginning of the T-section. The student may see it only when he takes a module for the second time, such as in a REVIEW or RECAP status. (Sample 22)

The only standard page in a P-section is the transition page, which is the starting point for a student who exercises the PRACTICE feature. There is no predetermined label for this page since the length of the T-section is variable. The other labeling conventions in P-sections have already been discussed under Titles. The last character of the master element label is generally A-E. The uncoded "F" page is used for documentation of response analysis. Labels that end with G through Z or 1-9 (except P) are generally indicative of slave elements.

Standardized pages are again used in Q-sections. Some are seen by the student and some are for documentation purposes only. Appendix E contains samples of the standardized pages described below:

Page 101[1] — This is the quiz-score computation page. There is one such page for each objective in a quiz and they are sequenced alphabetically (in the sixth position of the label). Two types of scores are shown on the page. One is the Q-score, or actual raw score, and the other is the IDM score, which is a standardized score used by the IDM to generate feedbacks and make sequencing decisions. (Sample 24)

The first line of this page is self-explanatory. The second is an abbreviated statement of the objective, followed by "scorable events"—the characteristics of the student's answer whose presence or absence indicates satisfactory performance on the objective. On the next line (labeled "events") the events are coded alphabetically. The "points" line indicates how many points each occurrence of an event is worth. Then the labels (CWIPs) which designate the questions that test the objective are listed. Beside each label and under the separate event and point headings is a number which indicates the number of times that event can occur at that label. A blank entry indicates zero occurrences at that label. The next column shows the maximum points that the student can earn, both for each question and for the objective. The A-score column shows the range of raw points for the objective and the points are grouped (on separate lines) to correspond to IDM scores. An IDM score of six is always the passing score. The continuation of the computation sheet contains other information needed by course managers and instructional logic coders. (Sample 25)

Page 102 — This is the transition page for the quiz. The student is reminded that options will not be available during the quiz and here is the last chance to use them until the quiz is finished. (Sample 26)

Page 103 — This page asks the student about his expectations on the level of achievement he will attain on the quiz. The Level of Aspiration (LOA) measure is taken in terms of IDM scores. Sample 27 shows the element used when a quiz tests only one objective. The second line (the one that begins with "your ability to:") is completed by displaying the abbreviated form of the objective (from the 101 page) as a roll-on text (ROT). Quizzes with multiple objectives have separate pages for each objective, such as 103A, 103B, etc.

Page 104 — This page is reserved for general directions to the student and a brief description of what the quiz is like. (Sample 28)

Page 106 — This page contains the response analysis for the quiz as a whole. The author defines the sequence of questions, under various conditions, for the second and "XX"[2] attempt at the quiz. (Sample 29)

The following standard pages use the CAT-ZAP labeling scheme, because the page numbers in a quiz will vary, based upon the number of questions. (Appendix E contains samples of these pages.)

Page CATOPI — On this page the student is given an opportunity to rate the module he has just completed. The same scale is used after every quiz in the course. (Sample 30)

---

[1] Q-section labels always have a number in the third position. A number other than 1 would indicate another version of the Q-section, but the page might be the same (e.g., page 201 may have the same content as page 101).

[2] XX = third attempt at a quiz. Proctor tutors student during quiz attempt.

Page CATPER — This page contains the performance summary for the student. Once again the abbreviated objective is displayed, and then the student's Level of Aspiration (LOA) is shown in both words and numbers. The student is then told whether he passed or failed and his IDM score (with associated verbal description) is shown below his LOA estimate for comparison. The types of error listed on Sample 31 are not visible to the student unless he committed one or more of them, in which case the specific relevant message is unmasked via the instructional logic.

Page CATQRI(2) — The Quiz Review Answer (QRA) page tells the student that he passed the quiz, but did not score perfectly, and that he is about to see the correct answer(s) to the question(s) he missed. (Sample 32)

Page CATQRA — This page contains the correct answer to the quiz question(s) missed on the previous try. (Sample 33)

Page CATEOQ — This page indicates the end of the quiz (EOQ) and prompts the student to enter comments, if desired. (Sample 34)

The instructional content and pedagogical techniques employed within a section are the domain of the course authors. The purpose of standardization is to ensure that the IDM can be correctly operationalized, non-authors can follow the logic and intent of a piece of instruction, the student has a sense of continuity of materials, instructional coders can correctly operationalize a module (division, course), and course administrators have adequate documentation available to properly manage the course when students are on-line.

VISUALS

The other materials in the courseware package were much less structured than the CRT text. All visuals were filmed in sequence of use, but standard frames were limited to the following:

(1) A title page for each module (content identical to Sample 9, Appendix E).

(2) Three "blank" frames between modules so that the film can more easily be spliced and thereby edited (Sample 35, Appendix E). The student sees these frames only if the P-scope has had a mechanical failure.

The only other documentation on the film (aside from the cue marks) was the frame label, located in the lower right-hand corner. The frames were referenced in the text with directions such as:

Look at P-SCOPE AAA99P.

For convenience and economy in course administration, the complete film was cut into filmstrips of visuals for one or more modules. Each strip was stored in a labeled canister for easy retrieval.

GLOSSARY

As mentioned earlier in the discussion of glossary documentation, the length of the term was determined by the system limitation of 32 characters in the title field. The amount of text for each glossary entry was limited by the size of the file record (700 characters) and by the size of the glossary block on the CRT screen (252 characters, in most cases).

Use of the entire record length of 700 characters was limited to entries for error messages. Requests for diagnostic codes occurred in only four modules of the course, so the size of the glossary block could be increased in these modules without decreasing the efficient use of available characters on the CRT screen in the course as a whole.

Hard-copy output for the glossary was programmed for sequencing of entries by terms and error codes in alphanumeric order. It was convenient, although not essential, to also store them in this fashion.

System programming in Project IMPACT made it necessary to input glossary entries in one file and transfer them to another file before they were available for student requests. This procedure made it possible to get "draft" regular FACS printouts of the latest changes before "final" dictionary-style printouts of the complete file.

STUDENT REFERENCE MANUAL

The manual was organized like a standard reference book, rather than by course structure. In general, the contents were grouped and sequenced in the order of presentation in the course. However, references needed only in specific places (such as quizzes) were generally included in the appendices, rather than in the main body of text.

The student was referred to the manual by topics, as well as by page numbers. To avoid possible confusion and unnecessary reference to information out of the instructional context, an index was not included. Instead, the student was encouraged to use a detailed table of contents to find the information needed. A copy of this table of contents is included in Appendix H of this document.

For convenience and economy in course administration, working copies of the manual for both students and staff were kept in standard loose-leaf notebooks so that pages or complete manuals could be replaced.

SUPPLIES

The availability of commercial programming supplies eliminated the problem of preparing student worksheets and other forms for the COBOL course. Coding and flowcharting materials were placed in the student stations, including pencils, coding forms, flowcharting templates, and flowcharting worksheets. Punch cards and other programming supplies were made available at the keypunch machines in the Student Ready Room and in the computer center.

Chapter 5

IMPLEMENTATION AND OUTCOMES

The innovative procedures described in this report have proven successful during an extensive operational phase of Project IMPACT. Standardization of design and documentation reduced the ambiguity of communications from author to programmer, thus facilitating the implementation stage for the courseware and reducing both time and effort required for transforming an author's specifications into reality.

Once implementation was completed, the debugging phase of the courseware preparation was begun. Activities during this phase were also enhanced by the management techniques embodied in the courseware subsystem. The majority of the problems identified by the "debuggers" could be solved by referencing on-line documentation, rather than disturbing the original author. The hard-copy version of this documentation then facilitated revision efforts because changes could be made off-line using exact replicas of the on-line material. This resulted in a more economical use of both personnel and computer time.

When the instructional programs reached an operational status, administrators relied heavily on the design and documentation to assure a smooth and accurate "running" of the courses. Standardization of form and location in the documentation allowed course administrators to quickly and easily diagnose and solve student problems by using the hard-copy course printouts (FACS) and the title listings.

Changes in the hardware/software subsystem were minor during this stage. However, because the courseware subsystem was an integrated package, instructional materials could usually be adapted to the software changes with little or no loss of time and dollars. The system's programmers could easily ascertain the effects of changes on the courseware by scanning the relevant documentation, and thereby making their changes in a manner which minimally affected existing materials. If a change was unavoidable, portions of the course to be affected were easily identified and appropriate action was taken to minimize unpredicted effects of the change.

Effective management of the courseware was one of the major outcomes of Project IMPACT during Cycle II of its effort. Since the other systems all interacted at the course level, this outcome is understandable. The innovative procedures employed rather common concepts of structure, documentation, and standardization of form, certainly not unique to any single computer application to educational courses. They are proven successful and are generalizable to any courseware system where efficient management of the instructional components is either required or desired.

The logic and content of the courseware subsystem of Project IMPACT may be difficult or even undesirable to replicate *in toto.* Replication in concept, however, would be valuable to any computer-based learning project.

Bibliography

DOCUMENTS ASSOCIATED WITH PROJECT IMPACT

"The Development and Maintenance of Optimal Learning Conditions", by Robert J. Seidel, paper for symposium at American Psychological Association convention, Washington, September 1967; issued under the title, *A General Systems Approach to the Development and Maintenance of Optimal Learning Conditions*, by Robert J. Seidel and Felix F. Kopstein, as Professional Paper 1-68, January 1968.

"Comment on Schurdak's 'An Approach to the Use of Computers in the Instructional Process and an Evaluation'," by Felix F. Kopstein and Robert J. Seidel, *American Educational Research Journal*, vol. 4, no. 4, November 1967.

"Discussion of a Unique Approach to CAI: Project IMPACT", by Robert J. Seidel, paper for USCONARC Training Innovations Conference, Fort Benning, Ga., September 1968; included in *Innovations for Training*, Professional Paper 6-69, February 1969.

"Product or Systems Research as Applied to Education for Business," by Felix F. Kopstein, paper for meeting of Research Institute, National Business Education Association, St. Louis, Mo., October 1968; *National Business Education Quarterly*, vol. 37, no. 3, Spring 1969; issued as Professional Paper 30-69, October 1969.

"Graph Theory as a Metalanguage of Communicable Knowledge," by Edward Kingsley, Felix F. Kopstein, and Robert J. Seidel, paper for annual meeting of the Society for General Systems Research, Dallas, Tex., December 1968; issued as Professional Paper 29-69, September 1969.

"Computers in Education: The Copernican Revolution in Education Systems," by Robert J. Seidel, paper for symposium at American Association for the Advancement of Science conference, Dallas, Tex., December 1968; *Computers and Automation*, March 1969; issued as Professional Paper 16-69, May 1969.

Project IMPACT: Computer-Administered Instruction Concepts and Initial Development, by Robert J. Seidel and the IMPACT Staff, Technical Report 69-3, March 1969.

"Is CAI Cost/Effective? The Right Question at the Wrong Time," by Robert J. Seidel, *Educational Technology*, vol. 9, no. 5, May 1969.

"Project IMPACT: Description of Learning and Prescription for Instruction," by Robert J. Seidel, Judy G. Compton, Felix F. Kopstein, Richard D. Rosenblatt, and Sally See, paper for Association for Computing Machinery symposium, Gaithersburg, Md., June 1969; issued as Professional Paper 22-69, June 1969.

"Rational vs. Empirical Approaches to Job/Task Descriptions for COBOL Programmers," by Felix F. Kopstein, paper for The Special Interest Group Computer Personnel Research of the Association for Computing Machinery annual conference, University of Chicago, June 1969; issued as Professional Paper 18-70; June 1970.

"The Computer as Adaptive Instructional Decision Maker," by Felix F. Kopstein and Robert J. Seidel, paper for International Symposium on Man-Machine Systems, Cambridge, England, September 1969; issued as Professional Paper 1-70, January 1970.

"Technology of Training: Project IMPACT," by J. D. Lyons, paper for CONARC briefing, Fort Monroe, Va., February 1970; included in *HumRRO Research in Training Technology*, Professional Paper 21-70, June 1970.

"Resource Allocations to Effect Operationally Useful CAI," by Robert J. Seidel and Felix F. Kopstein, paper for National Security Industrial Association (NSIA) conference, Washington, D.C., February 1970; issued as Professional Paper 12-70, April 1970.

"Psychology or Cybernetics as Basis for Instructional Strategy," by Felix F. Kopstein and Robert J. Seidel, paper for The American Educational Research Association, Minneapolis, Minn., March 1970; issued as Professional Paper 22-71, November 1971.

"CAI: Technological Misconceptions," letter by Robert J. Seidel, Felix F. Kopstein, and Ronald J. Swallow, *Science*, vol. 168, no. 3938, June 1970.

Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem, by the IMPACT Staff, Technical Report 70-22, December 1970.

"Theories and Strategies Related to Measurement in Individualized Instruction", by Robert J. Seidel, paper for The American Psychological Association Convention, Miami Beach, Fla., September 1970; issued as Professional Paper 2-71, March 1971. (Under joint sponsorship with National Science Foundation, Grant No. GJ 774)

Project IMPACT—Computer-Administered Instruction: Preparing and Managing the Content of Instruction, IMPACT Text-Handling Subsystem, by the IMPACT Staff, Technical Report 71-21, September 1971.

"Current Status of Computer-Administered Instruction Work Under Project IMPACT," by Robert J. Seidel, paper for CONARC Training Workshop, Fort Gordon, Ga., October 1971; issued as Professional Paper 18-72, July 1972.

Software Documentation Series. Project IMPACT—Computer-Administered Instruction: Functions for the Coursewriter III Language, by The IMPACT Staff, Research By-Product D1-71-2, June 1971.

Project IMPACT Software Documentation: Overview of the Computer-Administered Instruction System, by John Stelzer and Jean Garneau, Technical Report 72-21, August 1972.

Project IMPACT Software Documentation: II. The IMPACT Data Evaluation System—Version 2 (IDES-2), by Leslie Willis and John Stelzer, Research Product D1-72-1, August 1972.

Project IMPACT Software Documentation: III. The IMPACT Data Evaluation System—Version 1 (IDES-1), by John Stelzer and Leslie Willis, Research Product D1-72-2, August 1972.

Project IMPACT Software Documentation: IV. The Interface Subsystem Framework for Instructional Decision Modeling, by William Underhill and John Stelzer, Research Product D1-72-3, August 1972.

Project IMPACT Software Documentation: V. File Activity Control System (FACS), by Leslie Willis, Jean Garneau and John Stelzer, Research Product 51-72-4, August 1972.

Project IMPACT Software Documentation: VI. Volume 1, Zeus Functions and Design Concepts, by Jean Garneau and John Stelzer, Research Product D1-72-5, August 1972.

Project IMPACT Software Documentation: VI. Volume 2, Zeus Program Logis Descriptions, by Jean Garneau, William Underhill, and Doris Shuford, Research Product D1-72-5, August 1972.

Project IMPACT Software Documentation: VII. IMPACT's Computer-Administered Instruction Software Subsystem, Coursewriter III, and Its Functions, by Doris Shuford and John Stelzer, Research Product D1-72-6, August 1972.

Project IMPACT Software Documentation: VIII. Computer-Administered Instruction Computer Program Logic for COBOL2 Course of Instruction, by Douglas Spencer, Elizabeth Sowell, Leslie Willis and Jean Garneau, Research Product D1-72-7, August 1972.

Appendix A

STUDENT TASKS IN THE COBOL1 COURSE

Module 1 (Introduction to the COBOL Course)

By studying text and diagrams presented on the CRT and P-scope screens, and by working frequently interspersed practice items and questions, the student learns:

(1) The location and function of devices in the cubicle (student station).
(2) How to employ keyboard and light pen to—
  (a) Initiate and end instructional sessions
  (b) Compose, modify, and transmit answers
  (c) Make comments
  (d) Request glossary definitions.

Module 2 (Level 1 Pre-Test)

By means of multiple-choice, fill-in, and constructed-response questions, students are pre-tested on their ability to recognize, recall, and construct:

(1) File and record structures.
(2) Flowchart symbols.
(3) Programming logic.
(4) COBOL verbs and sentences (format and function).

Questions on these four areas are at the simplest level.

Module 3 (Level 1 Problem 1 Specifications)

Students are presented with simple program specifications (information needed to write the program) and are taught to identify within the specifications input (data given) and output (data to be generated by the program). Processing (operations to be performed on the input to obtain the output) is initially explained in terms of steps a clerk would take and then translated into analogous computer steps.

Students are required to recognize types of input and output devices and to identify input and output data items from specifications; when shown a set of processing steps, they are to decide if the steps are complete and determine how many output items are produced when specified steps are performed.

Module 4 (Level 1 Flowcharting 1)

A pre-test of flowcharting is first given to permit those with sufficient knowledge to skip the subsequent instructions. In the test, students are shown a flowchart with processing blocks and arrows giving the direction of flow; they are asked to identify the processes associated with specific block shapes and to predict the direction of flow and types of output occurring under specified circumstances.

The instruction proper is based upon the processing steps developed during Module 3. Students are shown a complete but unconnected set of processing blocks, corresponding to the steps necessary to produce one output item. With the aid of considerable cueing or "prompting," they are required to connect the blocks into an appropriate sequence. The critical flowchart techniques of "looping" and "end-of-file checking" are introduced and taught in a like manner; here, however, the student is further required to predict the output from a variety of correct and incorrect applications of these techniques to demonstrate his understanding of the function.

Following the instruction is a flowcharting post-test, similar in content to the pre-test, used to screen out students in need of remediation.

Module 5 (Level 1 Introduction to Procedure Division)

The general structure of a COBOL Procedure Division is derived from the flowchart written by the student in Module 4. The convention of using lines to indicate program flow is translated into:

(1) Paragraphs where flow is through a linear sequence of related steps.
(2) Paragraph names and written commands where the flow branches out of one linear sequence to another.

Students' ability to differentiate related groups of steps is measured by having them select meaningful names for sequences color-coded on the flowchart. The use of names for related items of data is introduced. When given examples of data items and appropriate names, the student is required to identify the class to which each item belongs.

Module 6 (Level 1 Coding the Procedure Division)

Referring to the flowchart developed in Module 4, and a complete set of specifications (including data-names), the student performs the following tasks: Given the blocks and flow lines of the flowchart one at a time and in order of execution and given detailed explanation of the function, format, and punctuation of the corresponding COBOL statement(s), the student is required to write the COBOL code one line at a time until the Procedure Division is complete.

Module 7 (Level 1 Review of Procedure Division)

Given a P-scope containing a complete Procedure Division, the student, by answering multiple-choice and fill-in questions, demonstrates his understanding of the format and function of the commands used in the Procedure Division.

Module 8 (Level 1 Criterion Test)

By means of multiple-choice, fill-in and constructed-response questions, students are pre-tested on their ability to recognize, recall and construct:

(1) File and record structures.
(2) Flowchart symbols.
(3) Programming logic.
(4) COBOL verbs and sentences (format and function).

Questions on these four areas are at the simplest level. These questions are parallel forms of those presented in Module 2.

Module 9 (Level 2 Pre-test)

Given part or all of specifications and/or flowcharts as required, the student by answering multiple-choice, fill-in, and constructed-response questions the student demonstrates:

(1) His understanding of the function and format of—
    (a) Flowcharting symbol shapes
    (b) COBOL verbs
    (c) Other COBOL reserved words
    (d) Data-names and procedures-names.

(2) The ability to arrange flowchart steps in an order appropriate to achieving output.
(3) The ability to predict individual flowchart steps from previous ones.
(4) The ability to code specified lines from the Procedure Division.
(5) The ability to code an entire Procedure Division (if he chooses to try).

Module 10 (Level 2 Problem 2 Specifications)

Presented with a one-page narrative specification describing the input, output, and processing requirement for Problem 2, the student answers questions. His answers to the multiple-choice and fill-in questions demonstrate his comprehension and interpretation of the program requirements.

Module 11 (Level 2 Flowcharting Problem 2)

Presented with the written specifications for Problem 2, and a series of appropriate flowchart symbols, the student writes the appropriate words or phrases for the flowchart steps shown (e.g., he writes "OPEN THE FILES" when asked what the next step should be). The student also demonstrates his knowledge of the mechanics of flowcharting by answering questions and adding flowchart connectors.

Module 12 (Level 2 Introduction to Record Description)

Given frames on the P-scope and a copy of the Level 1 problem, and with heavy prompting, the student is able to identify procedure-names, data names, and reserved words. Given the rules for construction of names, the student writes some of his own.

Module 13 (Level 2 Record Descriptions (renamed from Description of Data Division)

Given record descriptions from Program 1, record layouts for Program 2, and program specifications for Program 2 dealing with reformatting of input to output layouts, the student answers questions and performs sample on-line coding demonstrating his understanding of the following elements of a record description:

(1) Level-number
(2) Group and elementary items
(3) Data-name
(4) PICTURE clause
(5) Data type X
(6) Item length.

Module 14 (Level 2 Coding Data Division)

Given samples of correctly coded file and record descriptions and heavy prompting, the student answers multiple-choice and fill-in questions regarding the format and function of each line of coding; then, he codes from specifications, in order, an input file description, an input record description, an output file description, and an output record description.

Given explanations of the format and functions of elements in an independent item description with a VALUE clause, the student codes a complete (two-line) Working-Storage Section.

Module 15 (Level 2 Identification and Environment Divisions)

Given a printout of the Level 1 program the student is able to answer questions about the Identification Division. Given a printout on the P-scope of another program, he is able to write the two lines of coding for the Identification Division required by COBOL for the Level 2 problem. Given a P-scope showing formats for the above, the student copies the next 3 lines of a COBOL program required by this installation, substituting the current date and the student's own name.

Given a copy of the Environment Division for the Level 1 program and his own coded Data Division for the Level 2 problem, the student is able to write, with heavy prompting, the coding for the Environment Division for the Level 2 problem. The student then answers questions about punctuation, margins, and reserved words in the lines he has coded.

Module 16 (Level 2 Procedure Division Problem 2)

Given a printout of the Procedure Division for Program 1, his flowchart for Problem 2 and heavy prompting for each line of coding, the student writes the appropriate Procedure Division entries using:

(1) OPEN verb
(2) READ verb
(3) AT END clause
(4) GO TO
(5) IF
(6) Nonnumeric literals
(7) Correct syntax
(8) Qualified data-names
(9) Meaningful procedure-names composed by student.

Module 17 (Level 2 Procedure and Identification Division for Problem 2)

Given coding sheet specifications for Problem 2, and his flowchart for Problem 2, the student writes COBOL statements using MOVE, WRITE, IF, GO TO, DISPLAY, and CLOSE.

The student desk-checks his program, codes the Identification Division and answers questions on his logic and syntax.

Module 18 (Level 2 Post-Test)

The student answers multiple-choice and fill-in questions demonstrating his understanding of the function and/or usage of the following kinds of terms and concepts:

(1) COBOL terms
(2) Flowchart symbols.

Given sample problem specifications, and input and output record layouts, the student answers several questions dealing with the following:

(1) - Selection of correct descriptive statements regarding outputs of the program.
(2) - Arrangement of functional operations in proper sequence.
(3) - Insertion of processing steps in empty flowchart symbols.
(4) - Connection of flowchart symbols and their associated descriptions.

The student codes COBOL statements for Procedures and Data Divisions from functional specifications, then codes a complete Procedure Division from specifications, record layouts, and completed Identification, Environment, and Data Divisions (showing assigned data-names).

Appendix B

DESCRIPTION OF THE INTRO COURSE

The purpose of this course is to introduce the student to the features of the computer assisted instruction (CAI) environment. It is independent of any subsequent substantive course and serves only to familiarize the student with the parameters and mechanics of the CAI system in which he will learn. The student is "learning by doing" in that he is sent to the terminal and told to sit down, make himself comfortable and just "Press Send". From then on he is learning about the system, from the system. At the end of INTRO the student is "signed off" the course and told that he is now ready to take any substantive course he desires (or is prescribed).

There are presently two versions of the INTRO course. Both versions consist of a single division (I), within which modules (referred to as "topics" to the student) are serially ordered. The thirteen modules of the INTRO course are listed below:

IAA	(1)	INTRODUCTION
IBA	(2)	EQUIPMENT IN STUDENT STATION
ICA	(3)	HOUSEKEEPING TECHNIQUES[1]
IDA	(4)	USE OF THE SHIFT KEY, SPACE BAR, AND SPACE KEY
IEA	(5)	STUDENT OPINIONS
IFA	(6)	RESPONDING TO CAI QUESTIONS
IGA	(7)	USE OF THE HORIZONTAL TAB AND CR KEYS
IHA	(8)	USE OF THE REPEAT, VERTICAL TAB, AND HOME KEYS
IIA	(9)	USE OF THE TYPE, INSERT, AND DELETE KEYS
IJA	(10)	PRACTICE ON THE KEYBOARD
IKA	(11)	STUDENT OPTIONS
ILA	(12)	REQUESTING INFORMATION
IMA	(13)	RECORDING COMMENTS

The modules contain the standard A-, T-, P-, and Q-sections and full titles (disc number, label, CW?, etc.), but the rules for the structure and contents of the sections are not rigidly adhered to. For example, there are no formal objectives for INTRO since the goal of the course is only to acquaint the student with the overall system and thereby reduce spurious errors in later courses. Therefore the "objective" page is missing from INTRO A-sections (except IKB, ILB and IMB). Additionally, quizzes serve largely as tests for only the students' benefit and are neither scored (except right/wrong) nor do failures force a branch to remediation (as they do in a substantive course).

The first version of INTRO takes the student through all thirteen modules of the course in a tutorial mode. The characteristics of the CRT, manual, P-scope, student station, and the IDM features needed for later courses, are explained. Before the student enters his station, he is given a brief talk and is shown the computer center so that he is gradually introduced to the course environment.

The second version of INTRO replaces modules B, C, D, E, F, G, H, I, and J with an off-line lecture and demonstration of the CRT, manual, P-scope, and student station features. Topic A is left intact so that the student has a chance to practice what he has just been told. He can spend as much time as he wants filling lines with text and experimenting with the assorted keys on the keyboard. The alternate (or B) versions of modules K, L, and M are presented to the student as a single topic on course (IDM) features. Module L introduces the INFO- option and explains the inquiry (glossary) feature to the student. The student next sees Module M, which teaches him how to enter comments into the system via the CRT. Module K is divided into five "mini-modules" (term used for descriptive purposes only, not a valid term in the hierarchy of course structure) each of which teach the student about one of the five learner-control features of the IDM (Quiz, Review, Practice, Recap, and Route). The course administrator can define which of the "mini-modules" the student should see. There are 32 possible variations (2^5) of presentation for this module (Table B-1).

---

[1] Refers to "sign-on" and "sign-off" procedures.

The first version of INTRO is written for use on specific hardware (Sanders) and explains this hardware to the student. The second version is more flexible, since the on-line portion deals only with some keyboard practice and the features available to the student in the courseware subsystem. In either case, when the student is finished with INTRO, he is acquainted with his new learning environment and is ready to go on to any substantive course.

Table B-1

Variation on Module IKB

Variation	Mini-Modules on:				
	Route	Recap	Review	Quiz	Practice
1	Y	Y	Y	Y	N
2	Y	Y	Y	Y	Y
3	Y	Y	Y	N	N
4	Y	Y	Y	N	Y
5	Y	Y	N	Y	N
6	Y	Y	N	Y	Y
7	Y	Y	N	N	N
8	Y	Y	N	N	Y
9	Y	N	Y	Y	N
10	Y	N	Y	Y	Y
11	Y	N	Y	N	N
12	Y	N	Y	N	Y
13	Y	N	N	Y	N
14	Y	N	N	Y	Y
15	Y	N	N	N	N
16	Y	N	N	N	Y
17	N	Y	Y	Y	N
18	N	Y	Y	Y	Y
19	N	Y	Y	N	N
20	N	Y	Y	N	Y
21	N	Y	N	Y	N
22	N	Y	N	Y	Y
23	N	Y	N	N	N
24	N	Y	N	N	Y
25	N	N	Y	Y	N
26	N	N	Y	Y	Y
27	N	N	Y	N	N
28	N	N	Y	N	Y
29	N	N	N	Y	N
30	N	N	N	Y	Y
31	N	N	N	N	N
32	N	N	N	N	Y

WHERE: Y = seen by student.
N = not seen by student.

Appendix C

DESCRIPTION AND OBJECTIVES OF THE COBOL2 COURSE

The purpose of this course is to teach the student some of the basic concepts of data processing using the COmmon Business Oriented Language (COBOL) as a vehicle. Satisfactory completion of objectives of the course require that the student is knowledgeable in both general ADP concepts and specific COBOL applications of these concepts. The course consists of four divisions which contain a total of 33 different modules.

The module descriptions below are taken from the Student Reference Manual and are available to the student while he is taking COBOL2 (which is why "topic" is used instead of "module").

DIVISION A TOPIC DESCRIPTIONS

Topic M **About This Course**

This is an orientation to COBOL programming and how you will learn to do it in the course. You are shown what COBOL is and does. The purpose and structure of the course is also explained.

Topic A **A Simplified COBOL Computer**

The main parts of a computer are discussed. Instruction about the basic elements of how a computer processes data is provided.

Topic B **Moving Data Through the Computer**

You are taught how to give commands to list data records. Included are formats for READ, MOVE, and WRITE commands for moving records from the input file to the output file.

Topic C **Names for Data**

The concept and rules of COBOL reserved words are introduced. The rules for selection of programmer-supplied-names are provided. You practice making up your own valid data-names.

Topic D **COBOL Sentences and Their Formats**

This topic explains the rules and formats for writing valid COBOL sentences, commands, and clauses. You are given the formats, including punctuation and margin rules, for OPEN, READ, MOVE, WRITE, CLOSE and STOP RUN, and you practice writing the commands for a Procedure Division.

Topic E **Names for Procedures**

The concept of paragraphs within the Procedure Division is explained. The definition and use of procedure-names are provided so that you can make up your own and use them correctly.

DIVISION A TOPIC DESCRIPTIONS *(Cont.)*

Topic F **Branches and Loops**

The sequence of COBOL commands and methods of changing the sequence with GO TO statements are discussed. Loops and infinite loops are taught by having you trace the sequence of an infinite loop.

Topic G **Opening Files**

You are shown how to write a proper opening paragraph consisting of your own procedure-name and a correct OPEN statement. OPEN and its relation to reading the files is also explained.

Topic H **Closing Files**

This topic introduces you to the concept of closing the files. You insert in the Procedure Division a closing paragraph containing your own procedure-name, a CLOSE statement and a STOP RUN statement.

Topic I **What to do at the End**

Explanation on how to write a complete READ sentence is given. You supply the missing READ sentence in a Procedure Division using a valid procedure-name in the AT END clause. You also code a sequence that processes a file of three records and then stops the program.

Topic J **Coding a Procedure Division**

The use of the COBOL coding form is introduced including explanation on how to prepare the coding form for keypunching. You code the Procedure Division for Program A on the coding form.

Topic K **Correcting a COBOL Program**

In a gaming situation you are given practice in locating and correcting errors in the Procedure Division of a COBOL program.

Topic L **Debugging Program A**

With the aid of error messages you can request through the glossary and some instructions on debugging procedures, you debug Program A. When you find errors, you correct your coding by repunching the appropriate cards and then resubmit your program for COMPILE-AND-GO.

DIVISION B TOPIC DESCRIPTIONS

Topic A **Identification and Environment Divisions**

Supplied with formats, rules, and data-names, you code the Identification and Environment Divisions for Program B. Instruction about the SELECT clauses is also included.

DIVISION B TOPIC DESCRIPTIONS *(Cont.)*

Topic B **File Section and Record Descriptions**

The format and contents of the File Section of a COBOL program and descriptions of the records contained in the files are explained. You code input and output file and record descriptions. Record descriptions and their use in the READ-MOVE-WRITE sequence are also covered.

Topic C **Flowcharting Program A**

Flowcharting is introduced here. The purpose and contents of flowchart blocks and the relationship of blocks to COBOL statements are explained. The use of the flowcharting worksheet and template is also covered. You flowchart a Procedure Division and match flowchart symbols with COBOL verbs.

Topic D **Counters**

This topic teaches counters, a programming tool to count records. The ADD statement is also introduced, and you code one for your Program B.

Topic E **Working-Storage**

The concept of working-storage and work areas set up for counters are introduced. COBOL rules concerning level numbers and PICTURE, USAGE and VALUE clauses are discussed. You code an entire Working-Storage Section.

Topic F **Display**

The use and placement of the DISPLAY statement in a COBOL program is taught. You code a DISPLAY statement for your Program B.

Topic G **The Complete Data Division**

This is a summary of the components of the Data Division previously introduced, specifically the File Section and Working-Storage Section. Presented with assigned data-names, formats, and specifications, you code an entire Data Division.

Topic H **Error Game 2**

Again, you locate, correct, and replace errors in a COBOL program.

Topic I **Debugging Program B**

In this topic you debug Program B. Information on diagnostics can be retrieved via the glossary. General directions for the debugging procedure are presented. You correct any errors you find and resubmit your program for COMPILE-AND-GO.

DIVISION C TOPIC DESCRIPTIONS

Topic A **Record Layouts**

The subdivision of records into data fields and items and their rearrangements from input to output are explained. You practice moving data fields from input specifications to different layout specifications.

Topic B **Record Descriptions**

Given record layouts, you learn to code input and output record descriptions. You are also given explanation about elementary and group items, level numbers, and data-name qualifications in records and record descriptions.

Topic C **Move Statements**

The relations between garbage, FILLER, and SPACES is explained. You practice coding the proper set of MOVE statements to clear leftover garbage and then move data in the input record to the output record according to layout specifications.

Topic D **Coding Program C**

With access to the manual, data-names, record layouts, and program specifications, you code the complete Program C and submit it for keypunching. Your program deck is then assembled for a COMPILE-AND-GO attempt.

Topic F **Debugging Program C**

In this topic you debug Program C. You are given information on error messages received and access to directions for the debugging procedure. When errors are found, you correctly repunch appropriate cards and then resubmit your program for COMPILE-AND-GO.

DIVISION D TOPIC DESCRIPTIONS

Topic A **Input/Output Specifications**

Analysis of program specifications is taught. You learn to specify the content of different outputs required by the specifications, recognize the conditions for each different output to be produced, and identify components produced by several input records.

Topic B **Understanding Conditionals**

The function of conditional statements and their formats in standard, written English and in flowchart symbols are explained. Given specifications, you learn proper placement and content of decision blocks and program flow from the decision blocks in a program.

Topic C **Flowcharting Program D**

The logic for Program D is developed in a flowchart. You are given specifications, input and output record layouts, and a few sample input file records to process through your flowchart.

Topic D **Coding Conditionals**

In this topic you learn to translate conditional statements, expressed in standard English and flowchart symbols, into COBOL sentences. Given sample Procedure Divisions with conditionals and information concerning the input record contents, you predict the flow of the program.

Topic E **Coding Program D**

Given some additional information, you code Program D. You have access to the specifications and flowchart for Program D and input and output record layouts. After checking coding and keypunching, you submit your program deck for COMPILE-AND-GO.

Topic G **Debugging Program D**

Program D is debugged here. You have access to directions for debugging and receive information about error messages via the glossary. When errors are found, you correct the coding by repunching the cards and then resubmit your program for COMPILE-AND-GO.

To complete the course the student must successfully attempt each module quiz. The order in which he takes the quizzes can vary. Figure C-1 shows the prerequisite structure for COBOL2. It should be read from right to left, starting with module G in Division D and then to all prior modules up to module M in Division A. It can be verbalized to "before the student takes module _____he must have completed module(s)_____." For example, in Division B, before the student takes module H he must have completed modules A and G and F. The prerequisite structure shown was established for the COBOL2 course during its behavioral design phase and is based on an analysis of the relative nestings of behavioral objectives within others.

The path structure for COBOL2 can be derived from the prerequisite diagram, but it cannot be read directly from Figure C-1. Figure C-2 shows a portion of the path structure for Division B. It starts with the assumption that when the student saw the menu which gave him a choice between modules A, B, or C, he chose C.

In comparing Figures C-1 and C-2, it can be seen that modules become available to the student only after their prerequisites are satisfied. When, because of the prerequisites, there are no alternative modules for the student at a given point in the course, he goes directly to the module he is eligible to see, obviously without a menu. If, at a choice point, the student does not want to choose a module, he can default to a random choice by the computer. Regardless of which path the student takes, however, eventually all objectives must be encountered.

The behavioral objectives for the COBOL2 course are listed below. Objectives 37 and 38 are the terminal course objectives.

1. Given: 1. Explanation in narrative and summary form.
   2. Multiple-choice questions.
   3. Yes-No questions for self-evaluation.

   Then: The student will be able to
   1. Demonstrate recognition of the meaning of the content, purpose, and structure of the IMPACT course in COBOL programming.

   Constraints: None

COBOL2 Prerequisite Structure

COBOL Course

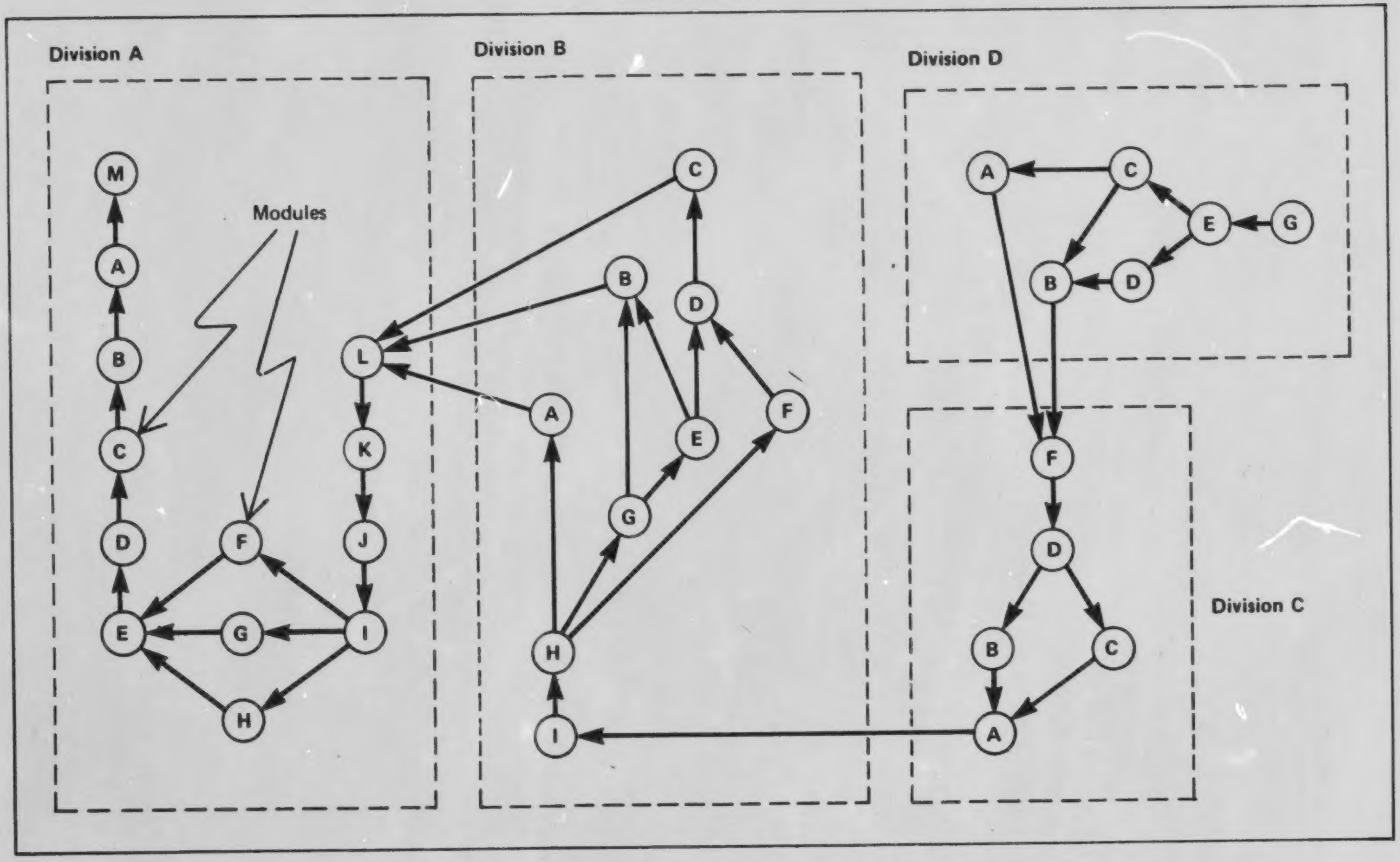
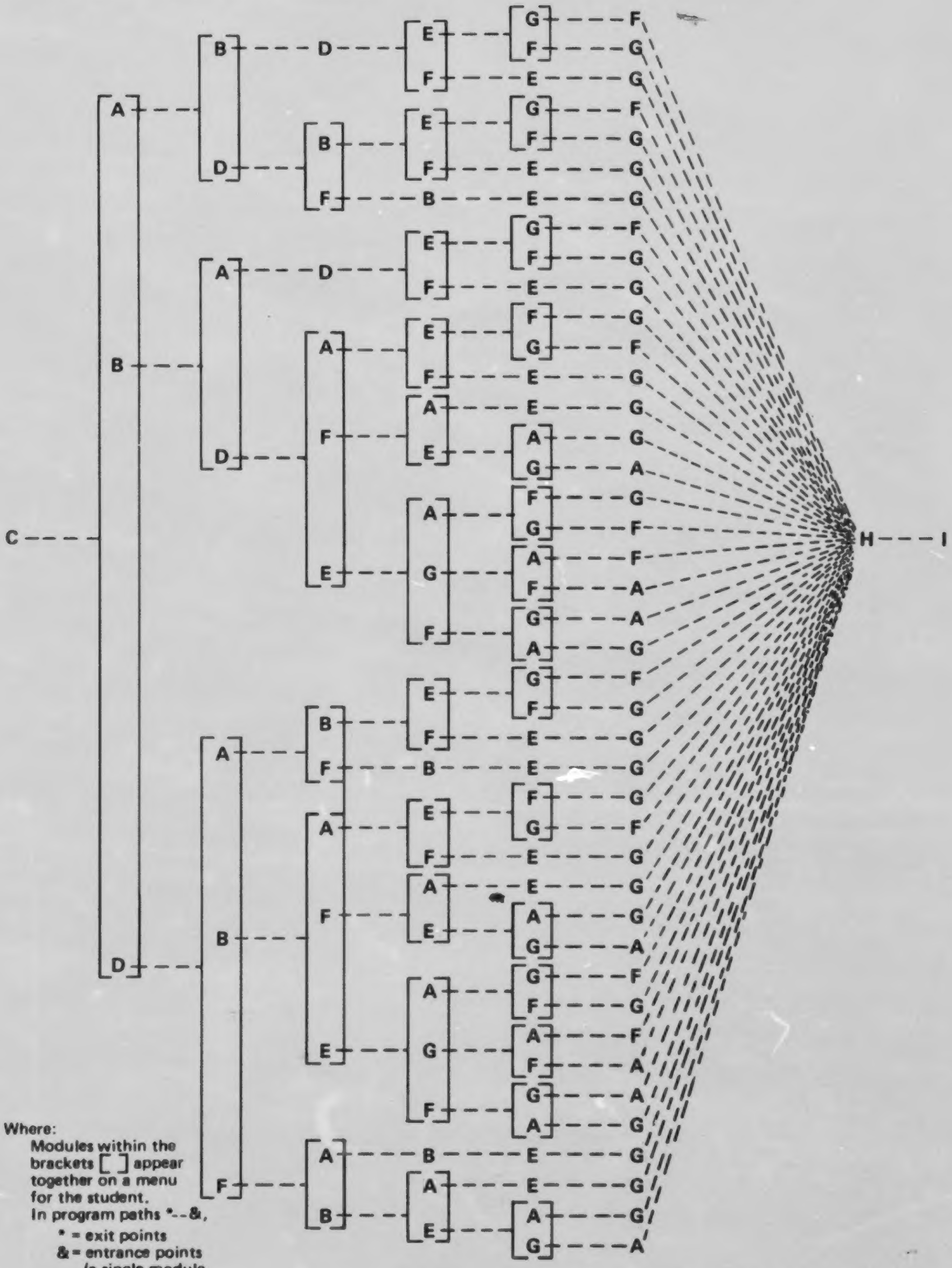

Division A
Modules
M
A
B
C
D
E
F
G
H
I
J
K
L
Division B
C
B
D
A
E
F
G
H
I
Division D
A
C
E
G
B
D
Division C
F
D
B
C
A

Figure C-1

Partial Path Diagram for Division B


G F
B D E F G
F E G
A G F
E
F G
B
D F E G
F B E G
G F
E
F G
A D
F E G
F G
E
A G F
B
F E G
A E G
F A G
D E
G A
F G
A
G F
C H I
A F
E G
F A
G A
F
A G
G F
E
F G
B
F E G
A
F B E G
F G
E
A G F
F E G
A E G
F A G
E
B
G A
D
G F
A
F G
A F
E G
F A
G A
F
A G
A B E G
A E G
F
B A G
E
G A

Where:
Modules within the brackets [ ] appear together on a menu for the student.
In program paths *--&,
* = exit points
& = entrance points (a single module or a menu).

Figure C-2

2. Given: A schematic representation of
    1. Input and output files in relation to the computer's input and output record areas.
  Then: The student will be able to
    1. Type the generic names for files and areas.
    2. Indicate their normal sequence.
    3. Identify the computer.
    4. Match generic names with real objects.
  Constraints: None

3. Given: A schematic representation of files and record areas as before
    1. A simplified format for statements.
    2. Data-names.
  Then: The student will be able to
    1. Type the correct sequence of READ, MOVE, and WRITE commands to move at least 2 records from the input file to the output file.
  Constraints: Allow 2 recoverable spelling mistakes.

4. Given: 1. List of COBOL reserved words.
    2. Rules for forming data-names.
  Then: The student will be able to
    1. Produce 6 different valid data-names.
  Given: The list of 12 suggested names.
  Then: The student will be able to
    1. Identify 6 out of 8 invalid names and 3 out of 4 valid names from a list of 12.
  Constraints: None.

5. Given: 1. COBOL formats for the following statements: OPEN, MOVE, WRITE, CLOSE, STOP RUN.
    2. Assigned data-names.
  Then: The student will be able to
    1. Type at least one COBOL statement for each format, beginning in the proper margin and ending with the proper punctuation.
  Constraints: None.

6. Given: 1. Simplified Procedure Division with No IF statements, double-spaced.
    2. Descriptions of the contents of 6 paragraphs to be formed by inserting procedure-names between the lines of code.
  Then: The student will be able to
    1. Generate and insert 6 valid procedure-names.
  Constraints: None.

7. Given: 1. Access to reference showing format for GO TO statements.
    2. A simple Procedure Division lacking procedure-names, GO TO statements, and conventional order of paragraphs-to-be.
    3. Specifications to achieve the following sequence of execution: OPEN-READ-MOVE-WRITE-CLOSE-STOP RUN.
  Then: The student will be able to
    1. Devise proper procedure-names.
    2. Devise proper GO TO statements.
    3. Insert them so as to meet specifications.
  Constraints: Dummy Procedure Division begins and ends with 2 empty lines; all statements within it are separated by 2 empty lines.

8. Given: A simple Procedure Division containing an infinite loop.
   Then: The student will be able to
   1. Identify the sequence of statements forming the infinite loop.

   Constraints: None.

9. Given:
   1. Format for OPEN statement.
   2. Data-names.
   3. A READ-MOVE-WRITE paragraph with 3 blank lines before and after.

   Then: The student will be able to insert a proper opening paragraph consisting of
   1. Student-supplied procedure-name.
   2. OPEN statement.

   Constraints:
   1. Program schema is same as for CLOSE statements (Objective 10).
   2. Student may use any 2 lines preceding the READ-MOVE-WRITE paragraph.
   3. Student must have been told that he cannot execute a READ statement until the file has been opened.

10. Given:
    1. Formats for CLOSE and STOP RUN statements.
    2. Data-names.
    3. A READ-MOVE-WRITE paragraph with 3 blank lines before and after.

    Then: The student will be able to insert a 3-line closing paragraph consisting of
    1. Student-supplied procedure-name.
    2. CLOSE statement.
    3. STOP RUN statement.

    Constraints: Program schema is same as for OPEN statements (Objective 9).

11. Given:
    1. Specifications for READ-MOVE-WRITE procedure corresponding to Division criterion problem.
    2. Data-names.
    3. Format for complete READ statement.
    4. Procedure Division (minus header) for program with missing READ statement.

    Then: The student will be able to
    1. Supply the missing READ statement using the correct procedure-name in the AT END clause.

    Constraints: None.

12. Given:
    1. Program completed by student for Objective 11.
    2. A hypothetical input file of 3 punched cards.

    Then: The student will be able to
    1. Predict the sequence in which lines of coding in the program will be executed as the input file is processed.

    Constraints: None.

13. Given:
    1. Narrative specifications for Division A criterion problem (code a simple Procedure Division).
    2. Data-names excepting procedure-names.
    3. Access to references containing formats, reserved words, rules for coding on a form, and a checklist.
    4. A sample of complete coding for a similar Procedure Division on a coding sheet.

Then: The student will be able to
   1. Code a complete Procedure Division on a COBOL form.
   2. Answer questions about his entries into name, identification and sequence fields.
   3. Answer questions about his coding and the areas (A and B) in which various lines appear.

Constraints: Student must code off-line into coding form.

14. Given:
   1. A printout of Procedure Division similar to that coded by the student (Objective 13) with numerous coding errors plus the resulting error messages.
   2. An interactive automaton (program) presenting successively an error message from the printout and a blank line for entering the corrected code.

   Then: The student will be able to
   1. Correct each successive line of coding until all errors have been eliminated.

   Constraints: Student must correct in 2 attempts or be given correct answers.

15. Given:
   1. Printout for COMPILE-AND-GO attempt based on student's coding (Objective 13).
   2. Corresponding card deck.
   3. A simplified error-message retrieval automaton (program) responding to a 4-digit code.
   4. Directions for debugging procedure.

   Then: The student will be able to
   1. Identify and correct coding errors on the printout and/or logic errors.
   2. Correctly repunch appropriate cards and insert them properly into his program deck.
   3. Submit deck for next COMPILE-AND-GO attempt.

   Constraints: Student debugs and resubmits program until it runs and produces proper output.

16. Given:
   1. Assigned input and output file-names.
   2. Assigned external-names for card-reader and printer.
   3. Rules for formation of program-names.
   4. Formats for Identification and Environment Divisions.

   Then: The student will be able to
   1. Code correct Identification and Environment Divisions.
   2. Recognize true statements about the functions of the SELECT clauses among incorrect ones.

   Constraints: None.

17. Given:
   1. Program specs to indicate that input records are moved "as is" to the output record area.
   2. Input and output record layouts—
      a. Input: Sample alphanumeric data of 80 characters.
      b. Output: Length and data not specified.
   3. Assigned file-names and record-names.
   4. Formats for file and record descriptions.

Then: The student will be able to

1. Correctly code 1 file description (FD) containing 1 record description.
2. Recognize true statements from among incorrect statements about—
   a. Various features of the FD.
   b. Various features of the record description.
   c. Relationships of record descriptions to record areas and READ-MOVE-WRITE statements.

Constraints: None

18. Given:

1. Properly coded dummy Procedure Division for Program A.
2. An explanation of relevant standard flowcharting symbols.
3. Program A verbs (including AT END, but not including GO TO).

Then: The student will be able to

1. Assign verbs to flowchart symbols, 1 symbol per verb.
2. Draw a Program A flowchart, and from it
3. Answer questions showing a sequence to—
   a. Print 2 records.
   b. Print last record, close file and stop run.

Constraints: None

19. Given:

1. Program A Procedure Division.
2. Access to reference containing ADD statement format.
3. Assigned counter-name.
4. Requirement to count records.

Then: The student will be able to

1. Insert proper ADD statement in 1 of 3 permissible locations within the program.
2. Predict the content of the counter as a file of 3 records is processed.

Constraints: Student must punch own ADD statement and properly insert card into his program deck.

20. Given:

1. Format for Working-Storage Section.
2. Assigned counter-name.
3. Narrative specifications requiring a counter for an input file of no more than 100 records.
4. Dummy File Section with coding for input and output files and records only.

Then: The student will be able to

1. Code a complete Working-Storage Section.
2. Indicate where it fits into the File Section.
3. Recognize correct from among incorrect statements about the functions of—
   a. 9 versus X in PICTURE clauses.
   b. The number within the parentheses.
   c. The VALUE clause.
   d. The USAGE clause.
   e. The Working-Storage Section.
   f. Relation of counters here to counter-name in ADD statement.

Constraints: None.

21. Given:
    1. Double-spaced Program A Procedure Division including ADD statement.
    2. Specifications to print number of records in the input file that have been counted together with a specified non-numeric literal.
    3. Format for DISPLAY statements including non-numeric literals.

    Then: The student will be able to
    1. Code the proper DISPLAY statement.
    2. Indicate the proper place for the DISPLAY statement in the Program A Procedure Division.

    Constraints: Student must punch own DISPLAY statement and properly insert card into his program deck.

22. Given:
    1. Assigned data-names.
    2. Access to references containing all pertinent formats as previously encountered (Topics B, D, E, F).
    3. Complete specifications for Program B (PRINT-AND-COUNT).

    Then: The student will be able to
    1. Code the appropriate complete Data Division.

    Constraints: None.

23. Given:
    1. Printout of a program corresponding to Program B (PRINT-AND-COUNT) with numerous coding errors plus the resulting error messages.
    2. An interactive automaton (program) presenting successively an error message from the printout and a blank line for entering corrected code.

    Then: The student will be able to
    1. Correct each successive line of coding until all errors have been eliminated.

    Constraints: Student must correct in 2 attempts or be given correct answer.

24. Given:
    1. Printout for COMPILE-AND-GO attempt based on student's coding for PRINT-AND-COUNT program.
    2. Corresponding card deck.
    3. A simplified error message retrieval automaton (program) responding to a 4-digit code.
    4. Access to directions for debugging procedure.

    Then: The student will be able to
    1. Identify and correct coding errors on the printout (also logic errors).
    2. Correctly repunch appropriate cards and insert them into his program deck.
    3. Submit deck for next COMPILE-AND-GO attempt.

    Constraints: Student debugs and resubmits program until it runs and produces proper output.

25. Given:
    1. 80-character input record layout containing several data fields as well as FILLER.
    2. Content of 1 sample record.
    3. Assigned data-names.
    4. Specifications for output record to be re-formatted.

    Then: The student will be able to
    1. Type-re-formatted sample record content, including spaces.

    Constraints: Width of data fields will be pre-marked.

26. Given: 1. Input record layout with numeric, alphabetic and alphanumeric fields as well as FILLER, also some group items.
2. Assigned data-names.
3. Specifications to re-format output.
Then: The student will be able to
1. Code complete input and output record descriptions.
Constraints: None.

27. Given: 1. Input and output record layouts with several fields (including group items, also FILLER).
2. Assigned data-names.
3. Specifications to move input records to output record area.
Then: The student will be able to
1. Code the proper set of MOVE statements on a coding sheet.
Constraints: Student must indicate when he believes all necessary MOVE statements have been coded.

28. Given: 1. Program specifications parallel to PRINT-AND-COUNT program (Program B) with requirement to re-format output record.
2. Input and output record layout.
3. Assigned data-names.
4. Access to reference manual containing formats, etc.
5. An interactive "supervisory" automaton (program) which asks for critical items of information and provides hints about the coding.
Then: The student will be able to
1. Code the complete Program C (PEOPLE-COUNT).
2. Submit it for keypunching plus COMPILE-AND-GO attempt.
Constraints: None.

29. Given: 1. Printout for COMPILE-AND-GO attempt based on student's coding for PEOPLE-COUNT program.
2. Corresponding card deck.
3. A simplified error message retrieval automaton (program) responding to a 4-digit code.
4. Access to directions for debugging procedure.
Then: The student will be able to
1. Identify and correct coding errors on the printout (also logic errors).
2. Correctly repunch appropriate cards and insert them into his program deck.
3. Submit deck for COMPILE-AND-GO attempt.
Constraints: Student debugs and resubmits program until it runs and produces proper output.

30. Given: 1. Specifications for Program D (PENSION-ELIGIBILITY).
2. Input and output record layouts.
3. Several sample records from the input file.
Then: The student will be able to
1. Specify content of the different outputs required by the specifications.
2. Recognize the necessary and sufficient conditions for each different output to be produced from among a set of plausible ones.
3. Identify output components produced by each of several sample input records.
Constraints: None.

31. Given: 1. Several conditional statements in ordinary language.
2. Corresponding equivalents in altered form, each with 2 non-equivalent statements parallel to this altered form.
Then: The student will be able to
1. Identify the correct equivalent of the initial statement.
Constraints: None.

32. Given: 1. Specifications for a program requiring 2 conditional decisions.
2. Flowchart for this program with decision blocks.
3. Several sample records from the input file matching none, some or all of the conditions.
Then: The student will be able to
1. Specify the sequence of steps executed in processing each of the sample records.
Constraints: None.

33. Given: 1. Specifications for a program requiring conditional decisions.
2. Flowchart for this program lacking 2 decision blocks.
Then: The student will be able to
1. Specify proper location for the decision blocks.
2. Specify content of the decision blocks.
3. Specify program flow from the decision blocks.
Constraints: None.

34. Given: 1. Specifications for Program D (PENSION-ELIGIBILITY).
2. Input and output record layouts.
3. A small file of records corresponding to the input file.
Then: The student will be able to
1. Prepare a proper flowchart.
2. Process the sample input data through the flowchart by stating the requisite sequence of verbs for each record.
Constraints: Student must use only these verbs: CLOSE, COUNT, DISPLAY, MOVE, OPEN, READ, START, START, STOP, WRITE.

35. Given: 1. A set of flowchart segments containing decision blocks.
2. Assigned data-names and procedure-names.
Then: The student will be able to
1. Write the appropriate IF statement for the corresponding decision block.
Constraints: None.

36. Given: 1. A Procedure Division containing several properly coded IF statements.
2. A short input file.
3. Appropriate condition indicators.
Then: The student will be able to
1. Specify the sequence of Procedure Division steps that will be executed as each record is processed.
Constraints: None.

37. Given: 1. Specifications for Program D (PENSION-ELIGIBILITY).
2. Input and output record layouts.
3. Printout for Program C (PEOPLE-COUNT).
4. Flowchart for Program D (from Topic C).
5. An interactive automaton (program) that calls attention to changes in Program C needed to match specifications for Program D and provides hints and coding.

Then: The student will be able to

1. Code the complete Program D (PENSION-ELIGIBILITY).
2. Desk check his program.
3. Submit it for keypunching plus COMPILE-AND-GO attempt.

Constraints: Desk check if done with Program C data cards.

38. Given:

1. Printout for COMPILE-AND-GO attempt based on student's coding for PENSION-ELIGIBILITY program.
2. Corresponding card deck.
3. Access to directions for debugging procedure.

Then: The student will be able to

1. Identify and correct coding errors on the printout (also logic errors).
2. Correctly re-punch appropriate cards and insert them into his program deck.
3. Submit deck for next COMPILE-AND-GO attempt.

Constraints: Student debugs and re-submits program until it runs and produces proper output.

Satisfactory performance on the objectives for Divisions A and B (1-15 and 16-24) constitute putting together a complete COBOL program in "bits and pieces." In Division A of the course the student prepared the Procedure Division of a program and in Division B of the course he finishes the program (Identification, Environment, and Data Divisions) and learns the bases of flowcharting. The Division C program requires the student to exhibit more transfer of knowledge, as there is less instructional support in Division C than in Divisions A and B. The Division D program is almost a completely indepent program, written by the student. It is an extension of the Division C program in that the program utilizes the same data, but the required outputs are different.

Specifications and sample programs for each division of the course can be found in Appendix D.

The COBOL2 course was not intended to satisfactorily teach all possible student populations. The entry requirements are not extensive, but experience has shown that an unqualified student wastes both his time and system time with little probability of success. The entry requirements are:

1. Vision adequate for reading CRT and film content and no physical impairment precluding the operation of the keyboard and light pen.
2. Ability to keypunch COBOL program cards accurately (no speed requirement), given simple directions.
3. Minimal score of 46 on the Aptitude Test for Programmer Personnel.
4. Minimal SCAT math score of 298 and verbal score of 291, or equivalent.
5. Vocabulary level of high school senior, or equivalent.

The requirements may be amended in a classroom environment, if students are allowed to take notes or do any outside study. This was not the case in Project IMPACT. Non-research applications could definitely allow the student memory and learning aids which an instructor feels would be helpful. The COBOL2 course was intended to be a total tutorial course capable of administering all requisite instructional on-line.

Appendix D

COBOL PROGRAMS USED IN COBOL2

PROGRAM A SPECIFICATIONS

Input: A file of records on punched cards.

Output: A printed list of all the data in the input records.

Prepare the PROCEDURE DIVISION of a COBOL program to produce the above output, assuming that the input and output records are the same length. Stop the run when all the records are processed.

The following names are assigned:

Program	CARD-TO-PAGE
Input file	DECK
Input record	INPUT-CARD
Output file	LISTING
Output record	LINE-OUT

Procedure-names are to be supplied by the programmer.

NOTE: The PROCEDURE DIVISION will be inserted by the proctor in a master deck that contains the rest of the program, the data records, and the JCL cards.

```
101010 IDENTIFICATION DIVISION.                                           A-MASTER
101020 PROGRAM-ID.                         CARD-TO-PAGE.                  A-MASTER

101030 AUTHOR.                             STUDENT-NAME.                  A-MASTER
101040 DATE-WRITTEN.                       OCTOBER 5,1971.                A-MASTER
101050 DATE-COMPILED. MAR 30,1972                                         A-MASTER
101060 REMARKS.                            PROGRAM A.                     A-MASTER

201010 ENVIRONMENT DIVISION.                                              A-MASTER
201020 CONFIGURATION SECTION.                                             A-MASTER
201030 SOURCE-COMPUTER.                    IBM-360-40.                    A-MASTER
201040 OBJECT-COMPUTER.                    IBM-360-40.                    A-MASTER
201050 INPUT-OUTPUT SECTION.                                              A-MASTER
201060 FILE-CONTROL.                                                      A-MASTER
201070     SELECT DECK,    ASSIGN TO UT-S-READER.                         A-MASTER
201080     SELECT LISTING, ASSIGN TO UT-S-PRINTER.                        A-MASTER

301010 DATA DIVISION.                                                     A-MASTER
301020 FILE SECTION.                                                      A-MASTER
301030 FD  DECK                                                           A-MASTER
301040     LABEL RECORD IS OMITTED,                                       A-MASTER
301050     DATA RECORD IS INPUT-CARD,                                     A-MASTER
301060     RECORD CONTAINS 80 CHARACTERS.                                 A-MASTER
301070 01  INPUT-CARD                      PICTURE IS X(80).              A-MASTER
301080 FD  LISTING                                                        A-MASTER
301090     LABEL RECORD IS OMITTED,                                       A-MASTER
301100     DATA RECORD IS LINE-OUT,                                       A-MASTER
301110     RECORD CONTAINS 80 CHARACTERS.                                 A-MASTER
301120 01  LINE-OUT                        PICTURE IS X(80).              A-MASTER
























401010 PROCEDURE DIVISION.                                                A-MASTER
401020 FIRST-THING.                                                       A-MASTER
401030     OPEN INPUT DECK, OUTPUT LISTING.                               A-MASTER
401040 NEXT-CARD.                                                         A-MASTER
401050     READ DECK, AT END GO TO FINISH.                                A-MASTER
401060     MOVE INPUT-CARD TO LINE-OUT.                                   A-MASTER
401070     WRITE LINE-OUT.                                                A-MASTER
401080     GO TO NEXT-CARD.                                               A-MASTER
401090 FINISH.                                                            A-MASTER
401100     CLOSE DECK, LISTING.                                           A-MASTER
401110     STOP RUN.                                                      A-MASTER
```

```
                       PROGRAM A TEST DATA
***** *   *  ***   ****     ***   ****    *   *  ***  *   * ****
  *   *   *   *   *          *   *         * *  *   * *   * *   *
  *   *****   *    ***       *    ***       *   *   * *   * ****
  *   *   *   *       *      *       *      *   *   * *   * *  *
  *   *   *  ***  ****      ***  ****       *    ***   ***  *   *



***** *****  **** *****     ***  *   * ***** ****  *   * *****
  *   *     *       *      *   * *   *   *   *   * *   *   *
  *   ****   ***    *      *   * *   *   *   ****  *   *   *
  *   *         *   *      *   * *   *   *   *     *   *   *
  *   *****  ***    *       ***   ***    *   *      ***    *
```

PROGRAM B SPECIFICATIONS

Input: A file of records on punched cards.

Output 1: A printed list of all the data in the input records.

Output 2: The total number of records processed is to be printed with this message:

number RECORDS WERE PROCESSED.

Prepare the IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, and DATA DIVISION of a COBOL program to produce the above output, assuming that the input and output 1 records are the same length. For the PROCEDURE DIVISION, use the cards from Program A, with the necessary insertions.

The following names are assigned:

Program	PRINT-AND-COUNT
Input file	DECK
Input record	INPUT-CARD
Output file	LISTING
Output record	LINE-OUT
Counter	LISTING-COUNTER

Procedure names are to be supplied by the programmer.

NOTE: The complete program deck will be assembled by the programmer. Data records and JCL cards will be added by the proctor.

```
101010 IDENTIFICATION DIVISION.                                        B-MASTER
101020 PROGRAM-ID.                  PRINT-AND-COUNT.                   B-MASTER

101030 AUTHOR.                      STUDENT-NAME.                      B-MASTER
101040 DATE-WRITTEN.                OCTOBER 5,1971.                    B-MASTER
101050 DATE-COMPILED. MAR 30,1972                                      B-MASTER
101060 REMARKS.                     PROGRAM B.                         B-MASTER

201010 ENVIRONMENT DIVISION.                                           B-MASTER
201020 CONFIGURATION SECTION.                                          B-MASTER
201030 SOURCE-COMPUTER.             IBM-360-40.                        B-MASTER
201040 OBJECT-COMPUTER.             IBM-360-40.                        B-MASTER
201050 INPUT-OUTPUT SECTION.                                           B-MASTER
201060 FILE-CONTROL.                                                   B-MASTER
201070     SELECT DECK,    ASSIGN TO UT-S-READER.                      B-MASTER
201080     SELECT LISTING, ASSIGN TO UT-S-PRINTER.                     B-MASTER

301010 DATA DIVISION.                                                  B-MASTER
301020 FILE SECTION.                                                   B-MASTER
301030 FD  DECK                                                        B-MASTER
301040     LABEL RECORD IS OMITTED,                                    B-MASTER
301050     DATA RECORD IS INPUT-CARD,                                  B-MASTER
301060     RECORD CONTAINS 80 CHARACTERS.                              B-MASTER
301070 01  INPUT-CARD                PICTURE IS X(80).                 B-MASTER
301080 FD  LISTING                                                     B-MASTER
301090     LABEL RECORD IS OMITTED,                                    B-MASTER
301100     DATA RECORD IS LINE-OUT,                                    B-MASTER
301110     RECORD CONTAINS 80 CHARACTERS.                              B-MASTER
301120 01  LINE-OUT                  PICTURE IS X(80).                 B-MASTER
301130 WORKING-STORAGE SECTION.                                        B-MASTER
301140 77  LISTING-COUNTER           PICTURE IS 999,                   B-MASTER
301150         USAGE IS COMPUTATIONAL, VALUE IS ZERO.                  B-MASTER
```

```
00053    401010 PROCEDURE DIVISION.                                           B-MASTER
00054    401020 FIRST-THING.                                                  B-MASTER
00055    401030     OPEN INPUT DECK, OUTPUT LISTING.                          B-MASTER
00056    401040 NEXT-CARD.                                                    B-MASTER
00057    401050     READ DECK, AT END GO TO FINISH.                           B-MASTER
00058    401055     ADD 1 TO LISTING-COUNTER.                                 B-MASTER
00059    401060     MOVE INPUT-CARD TO LINE-OUT.                              B-MASTER
00060    401070     WRITE LINE-OUT.                                           B-MASTER
00061    401080     GO TO NEXT-CARD.                                          B-MASTER
00062    401090 FINISH.                                                       B-MASTER
00063    401095     DISPLAY LISTING-COUNTER,                                  B-MASTER
00064    401096         " RECORDS WERE PROCESSED.".                           B-MASTER
00065    401100     CLOSE DECK, LISTING.                                      B-MASTER
00066    401110     STOP RUN.                                                 B-MASTER
```

```
  1                           PROGRAM B TEST DATA
  2  ***** *   * ***   ****    *** ****    *   *  ***  *   * ****
  3    *   *   *  *   *         *  *         * *  *   * *   * *   *
  4    *   *****  *    ***      *   ***       *   *   * *   * ****
  5    *   *   *  *       *     *      *      *   *   * *   * *  *
  6    *   *   * ***  ****     *** ****       *    ***   ***  *   *
  7
  8
  9
 10   ***** *****  **** *****     ***  *   *  ***** ****  *   * *****
 11     *   *     *       *      *   * *   *    *   *   * *   *   *
 12     *   ****   ***    *      *   * *   *    *   ****  *   *   *
 13     *   *         *   *      *   * *   *    *   *     *   *   *
 14     *   *****  ***    *       ***   ***     *   *      ***    *
 15
015 RECORDS WERE PROCESSED.
```

```
101010 IDENTIFICATION DIVISION.                                          C-MASTER
101020 PROGRAM-ID.                         PEOPLE-COUNT.                 C-MASTER
101030 AUTHOR.                             STUDENT-NAME.                 C-MASTER
101040 DATE-WRITTEN.                       OCTOBER 5, 1971.              C-MASTER
101050 DATE-COMPILED. MAR 30,1972                                        C-MASTER
101060 REMARKS.                            PROGRAM C.                    C-MASTER

201010 ENVIRONMENT DIVISION.                                             C-MASTER
201020 CONFIGURATION SECTION.                                            C-MASTER
201030 SOURCE-COMPUTER.                    IBM-360-40.                   C-MASTER
201040 OBJECT-COMPUTER.                    IBM-360-40.                   C-MASTER
201050 INPUT-OUTPUT SECTION.                                             C-MASTER
201060 FILE-CONTROL.                                                     C-MASTER
201070     SELECT PERSONNEL-FILE, ASSIGN TO UT-S-READER.                 C-MASTER
201080     SELECT PENSION-LIST,   ASSIGN TO UT-S-PRINTER.                C-MASTER

301010 DATA DIVISION.                                                    C-MASTER
301020 FILE SECTION.                                                     C-MASTER
301030 FD  PERSONNEL-FILE                                                C-MASTER
301040     LABEL RECORD IS OMITTED,                                      C-MASTER
301050     DATA RECORD IS PEOPLE-POOP,                                   C-MASTER
301060     RECORD CONTAINS 90 CHARACTERS.                                C-MASTER
301070 01  PEOPLE-POOP.                                                  C-MASTER
301080     02  SOC-SECURITY                PICTURE IS 9(9).              C-MASTER
301090     02  FILLER                      PICTURE IS X.                 C-MASTER
301100     02  NAME.                                                     C-MASTER
301110         03  FAMILY                  PICTURE IS X(20).             C-MASTER
301120         03  GIVEN                   PICTURE IS X(20).             C-MASTER
301130     02  LOCATION                    PICTURE IS X(5).              C-MASTER
301140     02  FILLER                      PICTURE IS X(21).             C-MASTER
301150     02  YEAR-HIRED                  PICTURE IS 9999.              C-MASTER
302010 FD  PENSION-LIST                                                  C-MASTER
302020     LABEL RECORD IS OMITTED,                                      C-MASTER
302030     DATA RECORD IS OLD-TIMER,                                     C-MASTER
302040     RECORD CONTAINS 98 CHARACTERS.                                C-MASTER
302050 01  OLD-TIMER.                                                    C-MASTER
302060     02  FILLER                      PICTURE IS X(15).             C-MASTER
302070     02  HANDLE.                                                   C-MASTER
302080         03  GIVEN                   PICTURE IS X(20).             C-MASTER
302090         03  FAMILY                  PICTURE IS X(20).             C-MASTER
302100     02  FILLER                      PICTURE IS X(5).              C-MASTER
302110     02  DIGS                        PICTURE IS X(5).              C-MASTER
302120         02  FILLER                  PICTURE IS X(10).             C-MASTER
302130     02  GOVT-NUMBER                 PICTURE IS 9(9).              C-MASTER
302140     02  FILLER                      PICTURE IS X(10).             C-MASTER
302150     02  STARTIN-YEAR                PICTURE IS 9(4).              C-MASTER
```

PROGRAM C SPECIFICATIONS

Input:	A file of records on punched cards, containing personnel data.
Output 1:	A printed list of all the data in the input records, rearranged as shown in the output record layout.
Output 2:	The total number of records processed is to be printed with this message:

number RECORDS WERE PROCESSED.

Prepare a complete COBOL program to produce the above output, using the data-names in the input and output record layouts provided on page C-3 of the Student Reference Manual.

The following names are assigned:

Program	PEOPLE-COUNT
Input file	PERSONNEL-FILE
Input record	PEOPLE-POOP
Output file	PENSION-LIST
Output record	OLD-TIMER
Counter	LISTING-COUNTER

Procedure names are to be supplied by the programmer.

NOTE: The complete program deck will be assembled by the programmer. Data records and JCL cards will be added by the proctor.

```
302160 WORKING-STORAGE SECTION.                                              C-MASTER
302170 77  LISTING-COUNTER              PICTURE IS 999,                      C-MASTER
302180          USAGE IS COMPUTATIONAL, VALUE IS ZERO.                       C-MASTER

401010 PROCEDURE DIVISION.                                                   C-MASTER
401020 SET-UP-SHOP.                                                          C-MASTER
401030     OPEN INPUT PERSONNEL-FILE, OUTPUT PENSION-LIST.                   C-MASTER
401040 NEXT-EMPLOYEE.                                                        C-MASTER
401050     READ PERSONNEL-FILE, AT END GO TO BITTER-END.                     C-MASTER
401060     MOVE SPACES TO OLD-TIMER.                                         C-MASTER
401070     MOVE GIVEN IN NAME TO GIVEN IN HANDLE.                            C-MASTER
401080     MOVE FAMILY IN NAME TO FAMILY IN HANDLE.                          C-MASTER
401090     MOVE LOCATION TO DIGS.                                            C-MASTER
401100     MOVE SOC-SECURITY TO GOVT-NUMBER.                                 C-MASTER
401110     MOVE YEAR-HIRED TO STARTIN-YEAR.                                  C-MASTER
401120     WRITE OLD-TIMER.                                                  C-MASTER
401130     ADD 1 TO LISTING-COUNTER.                                         C-MASTER
401140     GO TO NEXT-EMPLOYEE.                                              C-MASTER
401150 BITTER-END.                                                           C-MASTER
401160     DISPLAY LISTING-COUNTER, " RECORDS WERE PROCESSED.".              C-MASTER
401170     CLOSE PERSONNEL-FILE, PENSION-LIST.                               C-MASTER
401180     STOP RUN.                                                         C-MASTER
```

```
              ROBERT A            SMITH                A-114       123568302       1944
              ARDITH L            FISH                 D-610       114131312       1946
              YALVERTON A         REEVES               C-100       165032796       1948
              WILLIAM K           EDGERTON             A-351       145723056       1948
              JENNIFER M          CAVELLIERI           C-350       452301864       1968
              ARTHUR T            ROBINSON             A-114       203857941       1968
              RICHARD A           WHITE                C-141       356281079       1948
              JANET C             AYRES                A-224       277510364       1952
              HORACE L            SHEPERD              A-238       452110387       1948
              WALTER C            LAJMANOVICH          B-210       285331407       1948
              JUAN J              FARRE                A-330       378201469       1948
              ANGELA J            GRAMS                A-351       389720140       1948
              FREDERICK Y         ALLENBY              D-110       253749602       1944
              JOHN R              CALLAHAN             D-208       387530241       1948
              EMIL B              POLANSKI             D-610       306552417       1948
              BARBARA P           WATSON               C-552       398605347       1948
              ALICE K             LANGLEY              A-108       512048390       1947
              ERWIN M             ELDENFELTER          B-210       423810793       1948
              ROMEO E             MADDEN               C-350       886302714       1970
              CAROLYN B           LEWIS                C-411       468012239       1948
              ALEXANDER F         SINCLAIR             B-210       523807721       1946
              OLIVER R            BARRETT              D-210       520736419       1961
              GREGORY L           TIMBERLAKE           D-416       582107699       1947
              LAUREL P            WILLIAMS             B-034       114308592       1948
              JULIET V            TENNEY               D-112       692013746       1949
              KRISTA C            BUCKEY               B-430       785214933       1961
              GEORGE R            HILTON               B-312       802431973       1948
              GEORGE B            CELLA                C-500       966201476       1945
              ALBERT D            TUTORIELLO           B-210       974301852       1910
029 RECORDS WERE PROCESSED.
```

PROGRAM D SPECIFICATIONS

Input 1: A card file, containing data records for every employee of the Top Hopper Department Store. The file is organized by departments (toys, men's wear, etc.).

Input 2: Header cards have been inserted in the Input 1 card file to separate the data records for each department. These header cards contain blank spaces in columns 1-9.

Output 1: A printed list of personnel who are eligible for pensions during the current year. This means all the employees who were hired in 1948.

Output 2: A special message is to be printed for medical staff who become eligible for pensioning. This means any eligible employee whose location is "C-552". For each of these employees, the following message is to be displayed:

MEDICAL STAFF DUE TO RETIRE: last-name, first-name, SS number

Output 3: When the sixth header card is reached, the following message is to be displayed and the program is to be stopped in the usual manner:

END OF RUN. number EMPLOYEES ARE ELIGIBLE.

NOTE: In Outputs 2 and 3 above, literals are shown in capital letters and data in small letters.

See film frame DA102P for the record layouts.

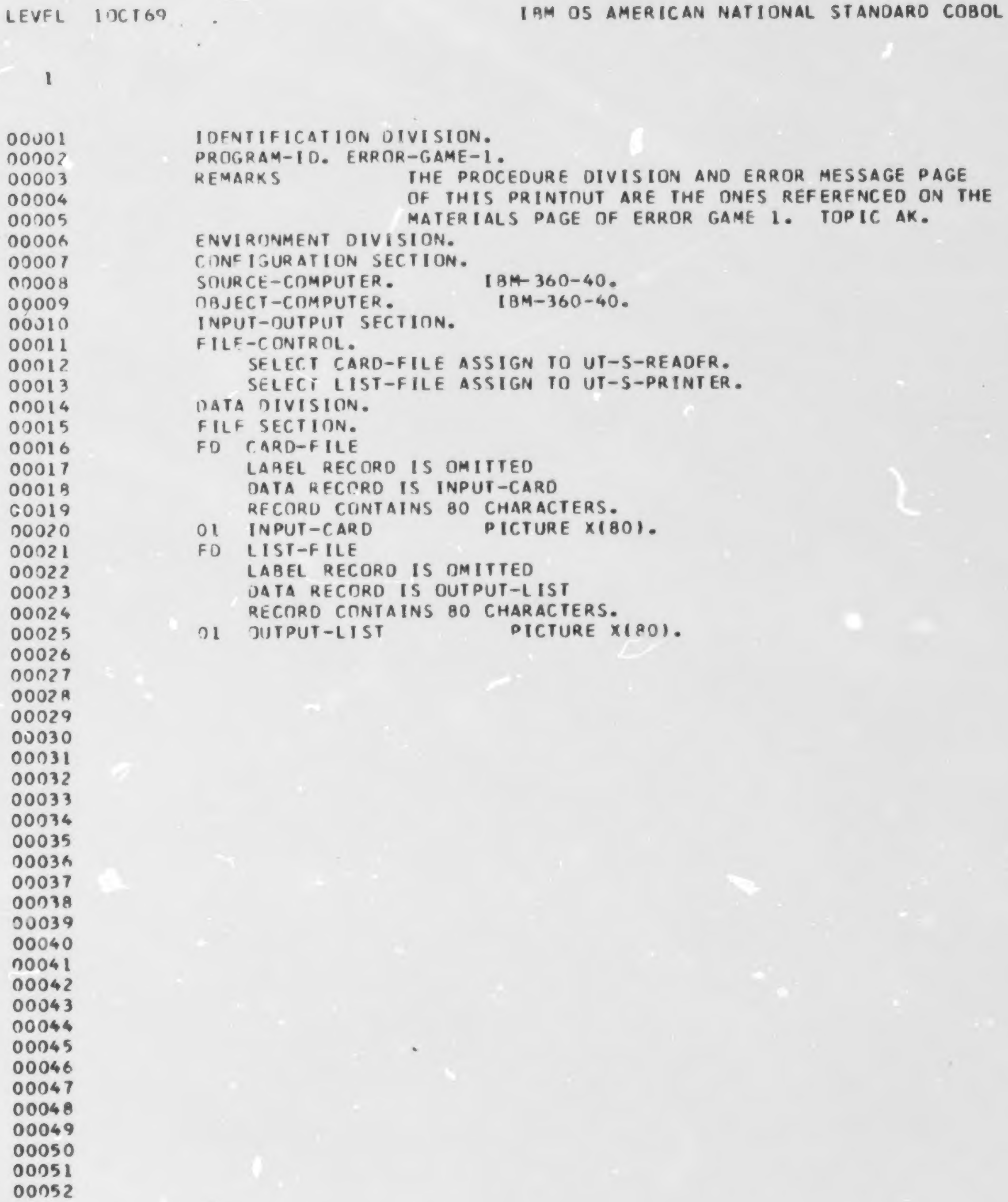
```
LEVEL  1OCT69                                      IBM OS AMERICAN NATIONAL STANDARD COBOL

   1

  101010 IDENTIFICATION DIVISION.                                                D-MASTER
  101020 PROGRAM-ID.               PENSION-ELIGIBILITY.                          D-MASTER
  101030 AUTHOR.                   STUDENT-NAME.                                 D-MASTER
  101040 DATE-WRITTEN.             OCTOBER 5, 1971.                              D-MASTER
  101050 DATE-COMPILED. MAR 30,1972                                              D-MASTER
  101060 REMARKS.                  PROGRAM D.                                    D-MASTER

  201010 ENVIRONMENT DIVISION.                                                   D-MASTER
  201020 CONFIGURATION SECTION.                                                  D-MASTER
  201030 SOURCE-COMPUTER.          IBM-360-40.                                   D-MASTER
  201040 OBJECT-COMPUTER.          IBM-360-40.                                   D-MASTER
  201050 INPUT-OUTPUT SECTION.                                                   D-MASTER
  201060 FILE-CONTROL.                                                           D-MASTER
  201070     SELECT PERSONNEL-FILE, ASSIGN TO UT-S-READER.                       D-MASTER
  201080     SELECT PENSION-LIST,   ASSIGN TO UT-S-PRINTER.                      D-MASTER

  301010 DATA DIVISION.                                                          D-MASTER
  301020 FILE SECTION.                                                           D-MASTER
  301030 FD  PERSONNEL-FILE                                                      D-MASTER
  301040     LABEL RECORD IS OMITTED,                                            D-MASTER
  301050     DATA RECORD IS PEOPLE-POOP,                                         D-MASTER
  301060     RECORD CONTAINS 80 CHARACTERS.                                      D-MASTER
  301070 01  PEOPLE-POOP.                                                        D-MASTER
  301080     02 SOC-SECURITY   PICTURE IS X(09).                                 D-MASTER
  301090     02 FILLER         PICTURE IS X.                                     D-MASTER
  301100     02 NAME.                                                            D-MASTER
  301110       03 FAMILY       PICTURE IS X(20).                                 D-MASTER
  301120     / 03 GIVEN        PICTURE IS X(20).                                 D-MASTER
  301130     02 LOCATION       PICTURE IS X(05).                                 D-MASTER
  301140     02 FILLER         PICTURE IS X(21).                                 D-MASTER
  301150     02 YEAR-HIRED     PICTURE IS 9999.                                  D-MASTER
  301160 FD  PENSION-LIST                                                        D-MASTER
  301170     LABEL RECORD IS OMITTED,                                            D-MASTER
  301180     DATA RECORD IS OLD-TIMER,                                           D-MASTER
  301190     RECORD CONTAINS 98 CHARACTERS.                                      D-MASTER
  301200 01  OLD-TIMER.                                                          D-MASTER
  301210 02  FILLER   PICTURE IS X(15).                                          D-MASTER
  301220     02 HANDLE.                                                          D-MASTER
  301230       03 GIVEN        PICTURE IS X(20).                                 D-MASTER
  301240       03 FAMILY       PICTURE IS X(20).                                 D-MASTER
  301250     02 FILLER         PICTURE IS X(05).                                 D-MASTER
  301260     02 DIGS           PICTURE IS X(05).                                 D-MASTER
  301270     02 FILLER         PICTURE IS X(10).                                 D-MASTER
  301280     02 GOVT-NUMBER    PICTURE IS X(09).                                 D-MASTER
  301290     02 FILLER         PICTURE IS X(10).                                 D-MASTER
  301300     02 STARTIN-YEAR   PICTURE IS 9(04).                                 D-MASTER
```

```
302010 WORKING-STORAGE SECTION.                                         D-MASTER
302020 77  LISTING-COUNTER   PICTURE IS 999,                            D-MASTER
302030         USAGE IS COMPUTATIONAL, VALUE IS ZERO.                   D-MASTER
302040 77  HEADER-COUNTER    PICTURE IS 9(3),                           D-MASTER
302050         USAGE IS COMPUTATIONAL, VALUE IS ZERO.                   D-MASTER

401010 PROCEDURE DIVISION.                                              D-MASTER
401020 SET-UP-SHOP.                                                     D-MASTER
401030     OPEN INPUT PERSONNEL-FILE, OUTPUT PENSION-LIST.              D-MASTER
401040 NEXT-EMPLOYEE.                                                   D-MASTER
401050     READ PERSONNEL-FILE, AT END GO TO BITTER-END.                D-MASTER
401060     IF SOC-SECURITY IS EQUAL TO SPACES, GO TO ONE-MORE-TIME,    D-MASTER
401070        ELSE NEXT SENTENCE.                                       D-MASTER
401080     IF YEAR-HIRED IS NOT EQUAL TO 1948, GO TO NEXT-EMPLOYEE,     D-MASTER
401090        ELSE NEXT SENTENCE.                                       D-MASTER
401100     MOVE SPACES TO OLD-TIMER.                                    D-MASTER
401110     MOVE GIVEN IN NAME TO GIVEN IN HANDLE.                       D-MASTER
401120     MOVE FAMILY IN NAME TO FAMILY IN HANDLE.                     D-MASTER
401130     MOVE LOCATION TO DIGS.                                       D-MASTER
401140     MOVE SOC-SECURITY TO GOVT-NUMBER.                            D-MASTER
401150     MOVE YEAR-HIRED TO STARTIN-YEAR.                             D-MASTER
401160     WRITE OLD-TIMER.                                             D-MASTER
401170     ADD 1 TO LISTING-COUNTER.                                    D-MASTER
401180     IF LOCATION IS EQUAL TO "C-552",                             D-MASTER
401190        DISPLAY "MEDICAL STAFF DUE TO RETIRE: ",                  D-MASTER
401200                NAME, SOC-SECURITY,                               D-MASTER
401210        ELSE NEXT SENTENCE.                                       D-MASTER
401220     GO TO NEXT-EMPLOYEE.                                         D-MASTER
401230 ONE-MORE-TIME.                                                   D-MASTER
401240     ADD 1 TO HEADER-COUNTER.                                     D-MASTER
401250     IF HEADER-COUNTER IS NOT EQUAL TO 6,                         D-MASTER
401260        GO TO NEXT-EMPLOYEE,                                      D-MASTER
401270        ELSE NEXT SENTENCE.                                       D-MASTER
402010 BITTER-END.                                                      D-MASTER
402020     DISPLAY "END OF RUN. ",                                      D-MASTER
402030             LISTING-COUNTER,                                     D-MASTER
402040            " EMPLOYEES ARE ELIGIBLE.".                           D-MASTER
402050     CLOSE PERSONNEL-FILE, PENSION-LIST.                          D-MASTER
402060     STOP RUN.                                                    D-MASTER
```

```
          ARDITH L            FISH                       D-610          114131312           1948
          YALVERTON A         REEVES                     C-100          165032796           1948
          WILLIAM K           EDGERTON                   A-351          145723056           1948
          RICHARD A           WHITE                      C-141          356291079           1948
          HORACE L            SHEPERD                    A-238          452110387           1948
          WALTER C            LAJMANOVICH                B-210          285331407           1948
          JUAN J              FARRE                      A-330          373201469           1948
          ANGELA J            GRAMS                      A-351          384720140           1948
          JOHN R              CALLAHAN                   D-209          887530241           1948
          EMIL B              POLANSKI                   D-610          306552417           1948
          BARBARA P           WATSON                     C-552          398605347           1948
          ERWIN M             ELDENFELTER                B-210          423810793           1948
          CAROLYN B           LEWIS                      C-411          463012239           1948
          LAUREL P            WILLIAMS                   B-034          114308592           1948
          GEORGE R            HILTON                     B-312          302431973           1948
MEDICAL STAFF DUE TO RETIRE: WATSON             BARBARA P           398605347
END OF RUN. 015 EMPLOYEES ARE ELIGIBLE.
```

ERROR GAME 1

Error Game 1 is a computer printout, similar to Program A, with no output data.



```
LEVEL  1OCT69                            IBM OS AMERICAN NATIONAL STANDARD COBOL

   1

00001          IDENTIFICATION DIVISION.
00002          PROGRAM-ID. ERROR-GAME-1.
00003          REMARKS            THE PROCEDURE DIVISION AND ERROR MESSAGE PAGE
00004                             OF THIS PRINTOUT ARE THE ONES REFERENCED ON THE
00005                             MATERIALS PAGE OF ERROR GAME 1.  TOPIC AK.
00006          ENVIRONMENT DIVISION.
00007          CONFIGURATION SECTION.
00008          SOURCE-COMPUTER.        IBM-360-40.
00009          OBJECT-COMPUTER.         IBM-360-40.
00010          INPUT-OUTPUT SECTION.
00011          FILE-CONTROL.
00012              SELECT CARD-FILE ASSIGN TO UT-S-READER.
00013              SELECT LIST-FILE ASSIGN TO UT-S-PRINTER.
00014          DATA DIVISION.
00015          FILE SECTION.
00016          FD  CARD-FILE
00017              LABEL RECORD IS OMITTED
00018              DATA RECORD IS INPUT-CARD
00019              RECORD CONTAINS 80 CHARACTERS.
00020          01  INPUT-CARD          PICTURE X(80).
00021          FD  LIST-FILE
00022              LABEL RECORD IS OMITTED
00023              DATA RECORD IS OUTPUT-LIST
00024              RECORD CONTAINS 80 CHARACTERS.
00025          01  OUTPUT-LIST           PICTURE X(80).
00026
00027
00028
00029
00030
00031
00032
00033
00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051
00052
```

```
  2

00053          PROCEDURE DIVISION.
00054          FIRST.
00055              OPEN INPUT CARD-FILE OUTPUT LIST-FILE.
00056          MAIN-ACTION
00057              READ CARD-FILE. AT END GG TO COMPLETION.
00058              MOVE INPUT-CARD TO OUTTPUT-LIST.
00059              WRITE OUTPUT-CARD.
00060              GO TO MAIN ACTION.
00061          COMPLETION.
00062              CLOSE CARD-FILE LIST-FILE.
00063              STOP RUN.
```

```
  3

*STATISTICS*     SOURCE RECORDS =    63     DATA DIVISION STATEMENTS =     4     PROCEDURE DIVISION STATEMENTS =     8
*OPTIONS IN EFFECT*     SIZE =   90000  BUF =     5032  LINECNT = 55  SPACE1, FLAGW,   SEQ,   SOURCE
*OPTIONS IN EFFECT*     NODMAP, NOPMAP, NOCLIST, NOSUPMAP, NOXREF,   LOAD, NODECK, QUOTE, NOTRUNC
```

```
  4

CARD   ERROR MESSAGE

54     IKF1087I-W     ' FIRST ' SHOULD NOT BEGIN A-MARGIN.
54     IKF1091I-W     FIRST IN A-MARGIN NOT VALID AS PROC-NM. ASSUME B-MARGIN.
53     IKF4003I-E     EXPECTING NEW STATEMENT. FOUND FIRST . DELETING TILL NEXT VERB OR PROCEDURE-NAME.
57     IKF1043I-W     END OF SENTENCE SHOULD PRECEDE READ . ASSUMED PRESENT.
57     IKF4070I-C     'AT-END' CLAUSE REQUIRED HERE. 'AT-END-NEXT-SENTENCE' ASSUMED.
57     IKF4003I-E     EXPECTING NEW STATEMENT. FOUND AT . DELETING TILL NEXT VERB OR PROCEDURE-NAME.
58     IKF3001I-E     OUTTPUT-LIST NOT DEFINED. DISCARDED.
59     IKF3001I-E     OUTPUT-CARD NOT DEFINED. STATEMENT DISCARDED.
60     IKF3001I-F     MAIN NOT DEFINED. STATEMENT DISCARDED.
60     IKF3001I-E     ACTION NOT DEFINED.
```

ERROR GAME 2

Error Game 2 is a computer printout, similar to Program B, with no output data.

```
LEVEL  1OCT69                               IBM OS AMERICAN NATIONAL STANDARD COBOL


   1


00001          IDENTIFICATION DIVISION.
00002          PROGRAM ID.                         STOCKS-AND-BONDS
00003          AUTHOR.                             J A DOE.
00004          DATE-WRITTEN.                       JULY 15, 1971.
00005          DATE-COMPILED. JUN  5,1972
00006          REMARKS              ALL FOUR DIVISIONS OF THIS PROGRAM ARE USED BY
00007                               THE STUDENT DURING ERROR GAME 2.
00008          ENVIRONMENT DIVISION.
00009          CONFIGURATION-SECTION.
00010          SOURCE-COMPUTER.                    IBM/360-40.
00011          OBJECT-COMPUTER.                    IBM/360-40.
00012          INPUT-OUTPUT SECTION.
00013          FILE-CONTROL-SECTION.
00014              SELECT INVESTMENTS, ASSIGN TO UT-S-READER.
00015              SELECT PROFIT,      ASSIGN TO UT-S-PRINTER.
00016
00017          DATA DIVISION.
00018          FILE SECTION.
00019          FD  INVESTMENTS.
00020              LABEL RECORD IS OMITTED,
00021              DATA RECORD IS STOCK-PRICE,
00022              RECORD CONTAINS 80 CHARACTERS.
00023          01  STOCK-PRICES                    PICTURE IS X(80).
00024          FD  PROFIT
00025              LABEL RECORD IS OMITTED,
00026              DATA RECORD IS OPTION,
00027              RECORD CONTAINS 80 CHARACTERS.
00028              OPTION                          PICTURE IS X(80).
00029          WORKING-STORAGE SECTION.
00030              TRANSACTION-COUNTER             PICTURE IS 999,
00031                    USAGE IS COMPUTATIONAL, VALUF IS ZERO.
00032
00033
00034
00035
00036
00037
00038
00039
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049
00050
00051
00052          PROCEDURE DIVISION.
```

```
  2

00053           CALL-BROKER.
00054               OPEN INPUT INVESTMENTS, OUTPUT PROFIT.
00055           SPEND-MONEY.
00056               READ INVESTMENTS, AT END GO TO CRASH.
00057               ADD 1 TO TRANSACTION-COUNTER.
00058               MOVE STOCK-PRICE TO OPTION.
00059               WRITE OPTION.
00060               GO TO SPEND-MONEY.
00061           CRASH.
00062               DISPLAY TRANSACTION-COUNTER,
00063                   "TRANSACTIONS TOOK PLACE BEFORE I LOST MY SHIRT."
00064               CLOSE INVESTMENTS, PROFIT.
00065               STOP RUN.
```

```
  3

*STATISTICS*      SOURCE RECORDS =    65     DATA DIVISION STATEMENTS =      3     PROCEDURE DIVISION STATEMENTS =    10
*OPTIONS IN EFFECT*     SIZE =   90000  BUF =     5032  LINECNT = 55  SPACE1, FLAGW,   SEQ,   SOURCE
*OPTIONS IN EFFECT*     NODMAP, NOPMAP, NOCLIST, NOSUPMAP, NOXREF,   LOAD, NODECK, QUOTE, NOTRUNC
```

```
  4

CARD    ERROR MESSAGE

2       IKF1087I-W     ' PROGRAM ' SHOULD NOT BEGIN A-MARGIN.
2       IKF1097I-E     PROGRAM-ID MISSING OR MISPLACED. IF PROGRAM-ID DOES NOT IMMEDIATELY FOLLOW
                       IDENTIFICATION DIVISION, IT WILL BE IGNORED.
9       IKF1087I-W     ' CONFIGURATION-SECTION ' SHOULD NOT BEGIN A-MARGIN.
9       IKF1004I-E     INVALID WORD CONFIGURATION-SECTION . SKIPPING TO NEXT RECOGNIZABLE WORD.
10      IKF1007I-W     SLASH NOT PRECEDED BY A SPACE. ASSUME SPACE.
10      IKF1002I-W     CONFIGURATION SECTION HEADER MISSING. ASSUMED PRESENT.
10      IKF1007I-W     360 NOT PRECEDED BY A SPACE. ASSUME SPACE.
10      IKF1043I-W     END OF SENTENCE SHOULD PRECEDE SLASH . ASSUMED PRESENT.
10      IKF1004I-E     INVALID WORD SLASH . SKIPPING TO NEXT RECOGNIZABLE WORD.
10      IKF1007I-W     -40 NOT PRECEDED BY A SPACE. ASSUME SPACE.
11      IKF1007I-W     SLASH NOT PRECEDED BY A SPACE. ASSUME SPACE.
11      IKF1007I-W     360 NOT PRECEDED BY A SPACE. ASSUME SPACE.
11      IKF1004I-E     INVALID WORD SLASH . SKIPPING TO NEXT RECOGNIZABLE WORD.
11      IKF1007I-W     -40 NOT PRECEDED BY A SPACE. ASSUME SPACE.
13      IKF1087I-W     ' FILE-CONTROL-SECTION ' SHOULD NOT BEGIN A-MARGIN.
13      IKF1004I-E     INVALID WORD FILE-CONTROL-SECTION . SKIPPING TO NEXT RECOGNIZABLE WORD.
14      IKF1003I-W     FILE-CONTROL PARAGRAPH NAME MISSING. ASSUMED PRESENT.
14      IKF2133I-W     LABEL RECORDS CLAUSE MISSING. DD CARD OPTION WILL BE TAKEN.
20      IKF1004I-E     INVALID WORD LABEL . SKIPPING TO NEXT RECOGNIZABLE WORD.
28      IKF1004I-E     INVALID WORD OPTION . SKIPPING TO NEXT RECOGNIZABLE WORD.
30      IKF1004I-E     INVALID WORD TRANSACTION-COUNTER . SKIPPING TO NEXT RECOGNIZABLE WORD.
15      IKF2183I-W     NO LEVEL 01 FOR FD OR SD.
15      IKF2146I-W     RECORD SIZE IN RECORD-CONTAINS CLAUSE DISAGREES WITH COMPUTED RECORD SIZE. USING
                       MAXIMUM COMPUTED SIZE.
57      IKF3001I-E     TRANSACTION-COUNTER NOT DEFINED. SUBSTITUTING TALLY .
58      IKF3001I-E     STOCK-PRICE NOT DEFINED. DISCARDED.
58      IKF3001I-E     OPTION NOT DEFINED.
59      IKF3001I-E     OPTION NOT DEFINED. STATEMENT DISCARDED.
62      IKF3001I-E     TRANSACTION-COUNTER NOT DEFINED. DISCARDED.
```

Appendix E

SAMPLE MATERIALS FROM THE INTRO AND COBOL2 COURSES

Sample
1 Manuscript Form
2 On-Line Documentation for a Hypothetical Practice Item
3 Excerpt From a Title Listing for CRT Text
4 Block Structure of CRT Text
5 Excerpt From a Title Listing for Visuals
6 FACS Printout of a Master Element
7 FACS Printout of a Slave Element
8 Printout of a Master Element on MINI-FACS
9 A-Section: Standard Page 00 for Titles
10 A-Section: Standard Page 01 for Prerequisites
11 A-Section: Standard Page 02 for Descriptions
12 A-Section: Standard Page 03 for Objectives
13 A-Section: Standard Page 04 for Directions
14 A-Section: Standard Page 04 for Materials (for Students)
15 A-Section: Standard Page 05 for Proctor Guide
16 A-Section: Standard Page 06 for Materials (for Proctors)
17 A-Section: Standard Page 07 for Response Analysis
18 A-Section: Supplement to 07 for Response Analysis
19 A-Section: Standard Page 08 for Key Terms
20 A-Section: Standard Page 09 for Special Terms
21 A-Section: Standard Page 09 for Special Terms (Quiz)
22 T-Section: Standard Page 10 for Transition
23 Title Listing of All Elements on a CRT Page
24 Q-Section: Standard Page 01 for Quiz Score Computations
25 Q-Section: Supplement to 01 for Quiz Score Computations
26 Q-Section: Standard Page 02 for Transition
27 Q-Section: Standard Page 03 for Level of Aspiration (LOA) Testing
28 Q-Section: Standard Page 04 for Quiz Directions
29 Q-Section: Standard Page 06 for Quiz Sequence
30 Q-Section: Standard Page for Student Opinions
31 Q-Section: Standard Page for Performance Evaluations
32 Q-Section: Standard Page for Quiz Review
33 Master Element Showing Correct Answers for Quiz Review
34 Q-Section: Standard Page for End of Quiz
35 Blank Frame Used to Separate Groups of Visuals

```
       HumRRO  CAI  Course  C O B O L 2   DIVISION_, MODULE_ _, VERSION_, SECTION_, PAGE_ _
LABEL '_ _ _ _ _ _, BLOCKS_ _ _ _ _, RESPONSE_ _ _, Fs_, Qs_, M_, P_, SEC_ - PUR_ _ _, CW_'
DISC   _ _ _ _ _ _, FORMAT_ _ _ _ _, P-SCOPE On Off Both GO TO _ _ _ _ _ _   FLOWCHART_________
      L   1   2   3   4   5   6   7   8   9   C   9   8   7   6   5   4   3   2   1   R
TABS  M...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...T...M/   PROCESS
_____________________________________________________________________________________________
  1                                                                                      /
  2                                                                                      /   Author  ______
  3                                                                                      /
  4                                                                                      /   Mss     ______
  5                                                                                      /
  6                                                                                      /   Date    ______
  7                                                                                      /
  8                                                                                      /
  9                                                                                      /   Review  ______
 10                                                                                      /
 11                                                                                      /
 12                                                                                      /   Edit 1  ______
 13                                                                                      /
 14                                                                                      /   Edit 2  ______
 15                                                                                      /
 16                                                                                      /   Type    ______
 17                                                                                      /
 18                                                                                      /
 19                                                                                      /   Review  ______
 20                                                                                      /
 21                                                                                      /
 22                                                                                      /   Create  ______
 23                                                                                      /
 24                                                                                      /   Modify  ______
 25                                                                                      /
 26                                                                                      /   Verify  ______
 27                                                                                      /
 28   bbb                                                                                /
 29   bbb                                                                                /   Titles  ______
 30   bbb                                                                                /
 31                                                                                          08-05-71  ljc
 32   INFO-
_____________________________________________________________________________________________
```

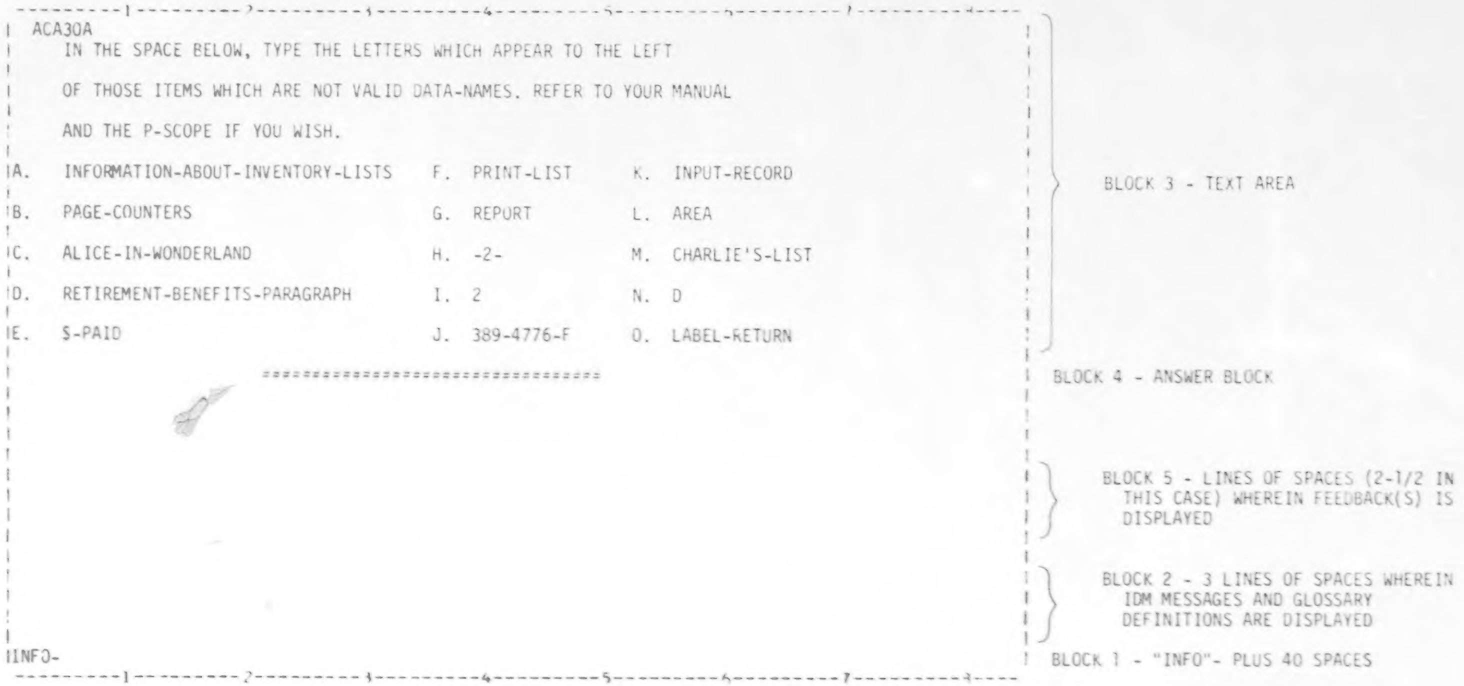
SAMPLE 1. Manuscript Form

```
CA   =  1
WA   =  2
WB1  =  3
IF CA GO TO CAF
IF TRY 1 WA GO TO TAF
IF TRY 1 WB1 GO TO TA1
IF TRY 1 ANYTHING ELSE GO TO TA1 THEN TA2
IF TRY2 WRONG GO TO WAF, AND ROT CA, BLOCK 5.
```

SAMPLE 2. On-Line Documentation for a Hypothetical Practice Item

DISC NO.	LABEL	CW?	SECTION	PURPOSE	FDBKS	P-SCOPE	GO TO	BLOCKING	RESPONSE
D02450	AAA38A	1	P	REC	3	S	AAA388-I	12345	BN4
F02469	AAA38F	0	P	RAF	0	0		00051	PSR

SAMPLE 3. Excerpt From a Title Listing for CRT Text



```
---------1---------2---------3---------4---------5---------6---------7---------8----
| ACA30A
|    IN THE SPACE BELOW, TYPE THE LETTERS WHICH APPEAR TO THE LEFT
|
|    OF THOSE ITEMS WHICH ARE NOT VALID DATA-NAMES. REFER TO YOUR MANUAL
|
|    AND THE P-SCOPE IF YOU WISH.
|
|A.  INFORMATION-ABOUT-INVENTORY-LISTS    F.  PRINT-LIST      K.  INPUT-RECORD
|
|B.  PAGE-COUNTERS                        G.  REPORT          L.  AREA
|
|C.  ALICE-IN-WONDERLAND                  H.  -2-             M.  CHARLIE'S-LIST
|
|D.  RETIREMENT-BENEFITS-PARAGRAPH        I.  2               N.  D
|
|E.  $-PAID                               J.  389-4776-F      O.  LABEL-RETURN
|
|                         ================================
|
|INFO-
---------1---------2---------3---------4---------5---------6---------7---------8----
```

BLOCK 3 - TEXT AREA

BLOCK 4 - ANSWER BLOCK

BLOCK 5 - LINES OF SPACES (2-1/2 IN THIS CASE) WHEREIN FEEDBACK(S) IS DISPLAYED

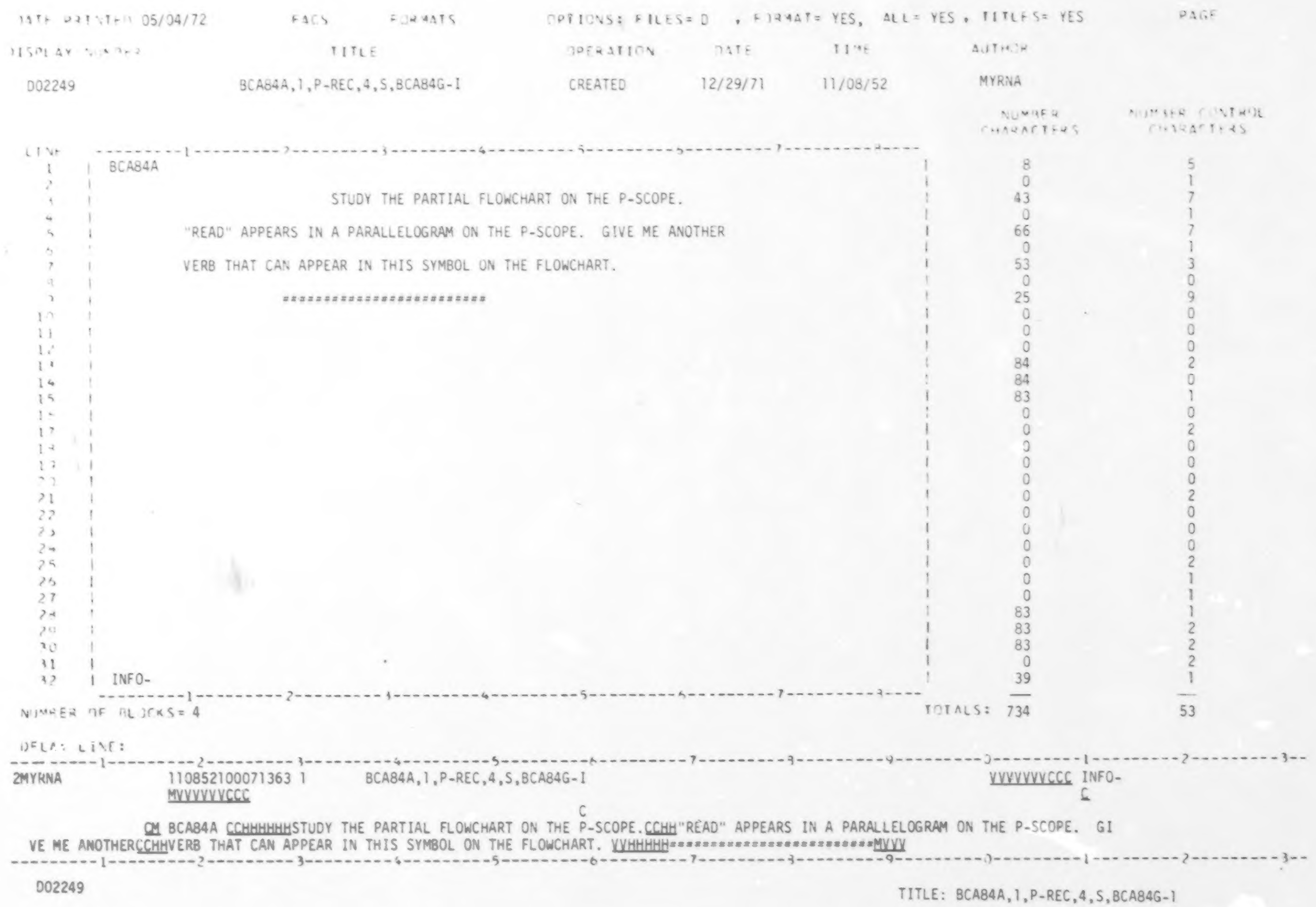
BLOCK 2 - 3 LINES OF SPACES WHEREIN IDM MESSAGES AND GLOSSARY DEFINITIONS ARE DISPLAYED

BLOCK 1 - "INFO"- PLUS 40 SPACES

SAMPLE 4. Block Structure of CRT Text

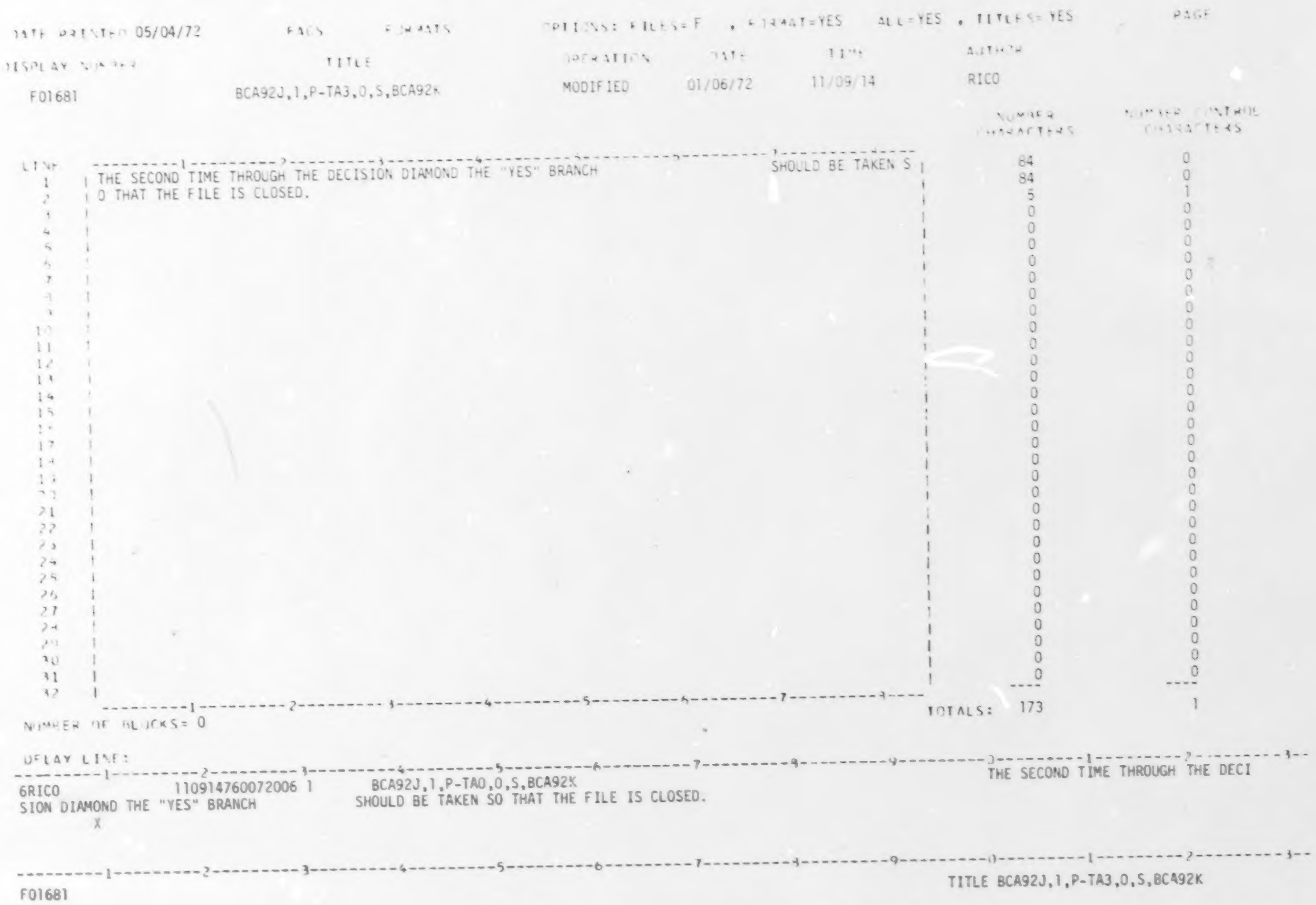
CUE MARK	P-SCOPE	ON	OFF	DESCRIPTION
P321	BDA30P	BDA22A	BDA22F	FLOWCHART-COUNTER
P325/567	ANIMAT	BDA22F	BDA24A	COUNTER ANIMATION

SAMPLE 5. Excerpt From a Title Listing for Visuals



```
DATE PRINTED 05/04/72        FACS      FORMATS        OPTIONS: FILES= D   , FORMAT= YES,  ALL= YES , TITLES= YES         PAGE

DISPLAY NUMBER                  TITLE                  OPERATION      DATE        TIME         AUTHOR

 D02249              BCA84A,1,P-REC,4,S,BCA84G-I        CREATED       12/29/71     11/08/52      MYRNA
                                                                                              NUMBER        NUMBER CONTROL
                                                                                            CHARACTERS        CHARACTERS
LINE    ---------1---------2---------3---------4---------5---------6---------7---------8----
  1   | BCA84A                                                                   |          8               5
  2   |                                                                          |          0               1
  3   |                     STUDY THE PARTIAL FLOWCHART ON THE P-SCOPE.          |         43               7
  4   |                                                                          |          0               1
  5   |        "READ" APPEARS IN A PARALLELOGRAM ON THE P-SCOPE.  GIVE ME ANOTHER  |         66               7
  6   |                                                                          |          0               1
  7   |        VERB THAT CAN APPEAR IN THIS SYMBOL ON THE FLOWCHART.              |         53               3
  8   |                                                                          |          0               0
  9   |                 =========================                                |         25               9
 10   |                                                                          |          0               0
 11   |                                                                          |          0               0
 12   |                                                                          |          0               0
 13   |                                                                          |         84               2
 14   |                                                                          |         84               0
 15   |                                                                          |         83               1
 16   |                                                                          |          0               0
 17   |                                                                          |          0               2
 18   |                                                                          |          0               0
 19   |                                                                          |          0               0
 20   |                                                                          |          0               0
 21   |                                                                          |          0               2
 22   |                                                                          |          0               0
 23   |                                                                          |          0               0
 24   |                                                                          |          0               0
 25   |                                                                          |          0               2
 26   |                                                                          |          0               1
 27   |                                                                          |          0               1
 28   |                                                                          |         83               1
 29   |                                                                          |         83               2
 30   |                                                                          |         83               2
 31   |                                                                          |          0               2
 32   | INFO-                                                                    |         39               1
        ---------1---------2---------3---------4---------5---------6---------7---------8----          ---             ---
NUMBER OF BLOCKS= 4                                                             TOTALS:  734              53

DELAY LINE:
---------1---------2---------3---------4---------5---------6---------7---------8---------9---------0---------1---------2---------3--
2MYRNA          110852100071363 1       BCA84A,1,P-REC,4,S,BCA84G-I                                      VVVVVVVCCC INFO-
                MVVVVVVCCC                                                                                         C
                                                            C
            CM BCA84A CCHHHHHHSTUDY THE PARTIAL FLOWCHART ON THE P-SCOPE.CCHH"READ" APPEARS IN A PARALLELOGRAM ON THE P-SCOPE.  GI
 VE ME ANOTHERCCHHVERB THAT CAN APPEAR IN THIS SYMBOL ON THE FLOWCHART. VVHHHHH=========================MVVV
---------1---------2---------3---------4---------5---------6---------7---------8---------9---------0---------1---------2---------3--

 D02249                                                                                      TITLE: BCA84A,1,P-REC,4,S,BCA84G-1
```

SAMPLE 6. FACS Printout of a Master Element

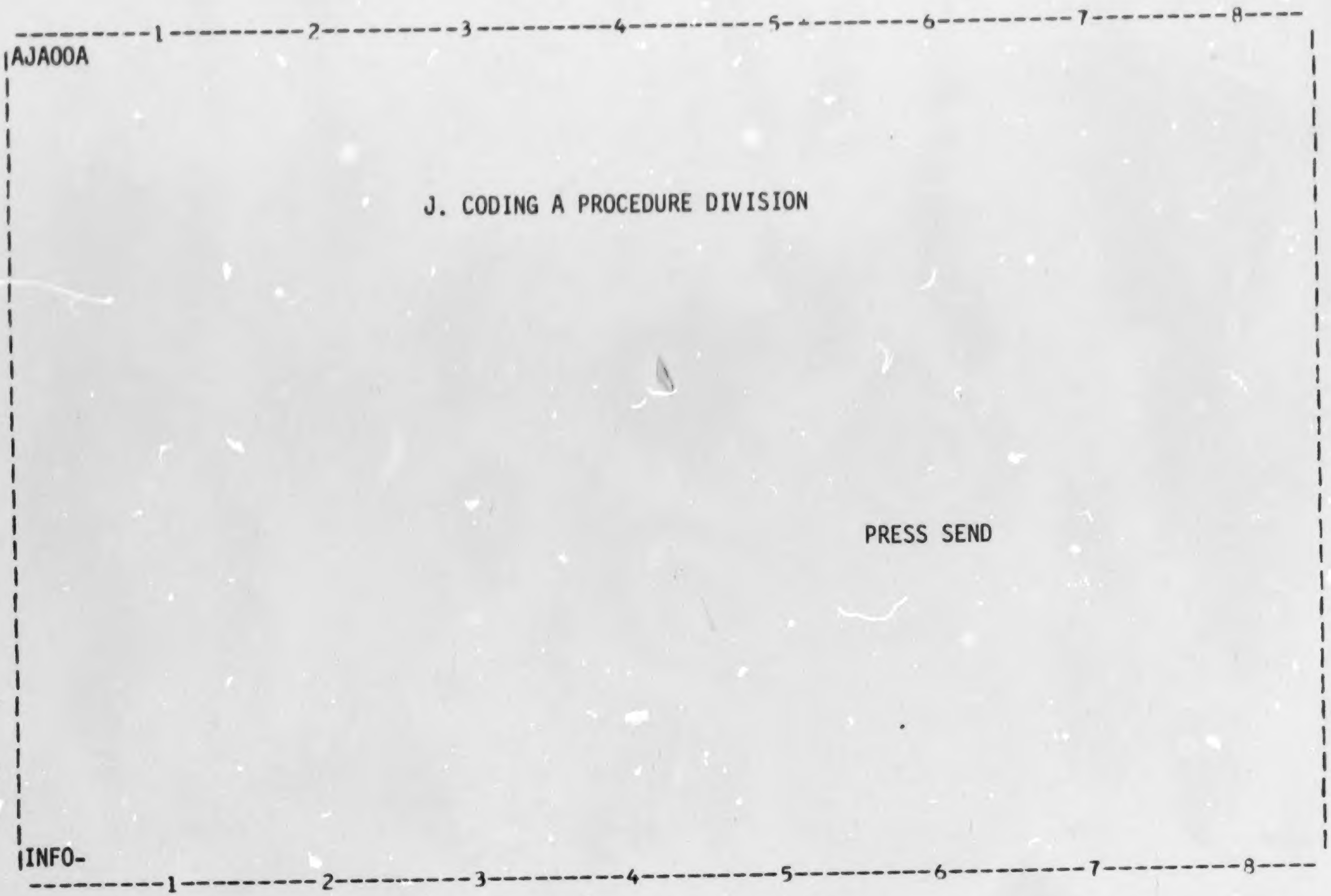


```
DATE PRINTED 05/04/72        FACS      FORMATS           OPTIONS: FILES=F   , FORMAT=YES   ALL=YES , TITLES=YES          PAGE

DISPLAY NUMBER                       TITLE               OPERATION      DATE        TIME        AUTHOR

  F01681              BCA92J,1,P-TA3,0,S,BCA92K          MODIFIED     01/06/72    11/09/14      RICO
                                                                                               NUMBER          NUMBER CONTROL
                                                                                              CHARACTERS         CHARACTERS

 LINE   ---------1---------2---------3---------4---------5---------6---------7---------8----
   1   | THE SECOND TIME THROUGH THE DECISION DIAMOND THE "YES" BRANCH     SHOULD BE TAKEN S |       84                 0
   2   | O THAT THE FILE IS CLOSED.                                                          |       84                 0
   3   |                                                                                     |        5                 1
   4   |                                                                                     |        0                 0
   5   |                                                                                     |        0                 0
   6   |                                                                                     |        0                 0
   7   |                                                                                     |        0                 0
   8   |                                                                                     |        0                 0
   9   |                                                                                     |        0                 0
  10   |                                                                                     |        0                 0
  11   |                                                                                     |        0                 0
  12   |                                                                                     |        0                 0
  13   |                                                                                     |        0                 0
  14   |                                                                                     |        0                 0
  15   |                                                                                     |        0                 0
  16   |                                                                                     |        0                 0
  17   |                                                                                     |        0                 0
  18   |                                                                                     |        0                 0
  19   |                                                                                     |        0                 0
  20   |                                                                                     |        0                 0
  21   |                                                                                     |        0                 0
  22   |                                                                                     |        0                 0
  23   |                                                                                     |        0                 0
  24   |                                                                                     |        0                 0
  25   |                                                                                     |        0                 0
  26   |                                                                                     |        0                 0
  27   |                                                                                     |        0                 0
  28   |                                                                                     |        0                 0
  29   |                                                                                     |        0                 0
  30   |                                                                                     |        0                 0
  31   |                                                                                     |        0                 0
  32   |                                                                                     |       ----              ----
        ---------1---------2---------3---------4---------5---------6---------7---------8---- TOTALS:  173                 1
NUMBER OF BLOCKS= 0

DELAY LINE:
---------1---------2---------3---------4---------5---------6---------7---------8---------9---------0---------1---------2---------3--
6RICO              110914760072006 1       BCA92J,1,P-TA0,0,S,BCA92K                                    THE SECOND TIME THROUGH THE DECI
SION DIAMOND THE "YES" BRANCH             SHOULD BE TAKEN SO THAT THE FILE IS CLOSED.
       X

---------1---------2---------3---------4---------5---------6---------7---------8---------9---------0---------1---------2---------3--
F01681                                                                                         TITLE BCA92J,1,P-TA3,0,S,BCA92K
```

SAMPLE 7. FACS Printout of a Slave Element

```
IAA24A

     WHEN YOU PRESS THE "SEND BLOCK" KEY YOU ARE "TALKING" WITH ME.
     YOU ARE TELLING ME THAT YOU ARE READY TO GO ON TO SOMETHING
     NEW. AT FIRST I'LL REMIND YOU TO PRESS THE "SEND BLOCK" KEY
     BY SAYING "PRESS SEND". LATER I'LL SHORTEN THIS TO "PS".
     AFTER AWHILE, YOU WILL NOT BE REMINDED. THEN I'LL ONLY TELL
     YOU TO "PRESS SEND" OR "PS" IN SPECIAL CASES.
         IF YOU UNDERSTAND WHAT I JUST EXPLAINED, "PRESS SEND".
         IF NOT, CALL THE PROCTOR.



INFO-

--------------------------------------------------------------------
INTRO               9 MAY 1973         IAA24A,1,T-EXP,0,0,IAA26A
```

SAMPLE 8. Printout of a Master Element on MINI-FACS



```
AJA00A

          J. CODING A PROCEDURE DIVISION



                                   PRESS SEND



INFO-
```

SAMPLE 9. A-Section: Standard Page 00 for Titles

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|AJA01A                                                                            |
|                                                                                  |
|                                    PREREQUISITES                                 |
|                                                                                  |
|     HERE IS A LIST OF THE TOPICS YOU HAVE COMPLETED.                             |
|                                                                                  |
|         A.  A SIMPLIFIED COBOL COMPUTER                                          |
|                                                                                  |
|         B.  MOVING DATA THROUGH THE COMPUTER                                     |
|                                                                                  |
|         C.  NAMES FOR DATA                                                       |
|                                                                                  |
|         D.  COBOL SENTENCES AND THEIR FORMATS                                    |
|                                                                                  |
|         E.  NAMES FOR PROCEDURES                                                 |
|                                                                                  |
|         F.  BRANCHES AND LOOPS                                                   |
|                                                                                  |
|         G.  OPENING FILES                                                        |
|                                                                                  |
|         H.  CLOSING FILES                                                        |
|                                                                                  |
|         I.  WHAT TO DO AT THE END                                                |
|                                                                                  |
|                                                                                  |
|                                                                                  |
|                                                                                  |
|INFO-                                                                             |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 10. A-Section: Standard Page 01 for Prerequisites

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|AJA02A                                                                              |
|                                                                                    |
|                                    DESCRIPTION                                     |
|        INTRODUCES THE USE OF THE COBOL CODING FORM.                                |
|           THE STUDENT FIRST SEES THE FORM IN TOPIC D.                              |
|        THE T, P, AND Q SECTIONS END WITH EXPLANATION AND DIRECTIONS                |
|        PREPARING CODING SHEETS FOR KEYPUNCHING.                                    |
|           THE T SECTION ALSO COVERS THE COMPLETE A-MASTER DECK THAT                |
|           THE STUDENT RECEIVES BACK.                                               |
|                                                                                    |
|                                                                                    |
|                                                                                    |
|                                                                                    |
|INFO-                                                                               |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

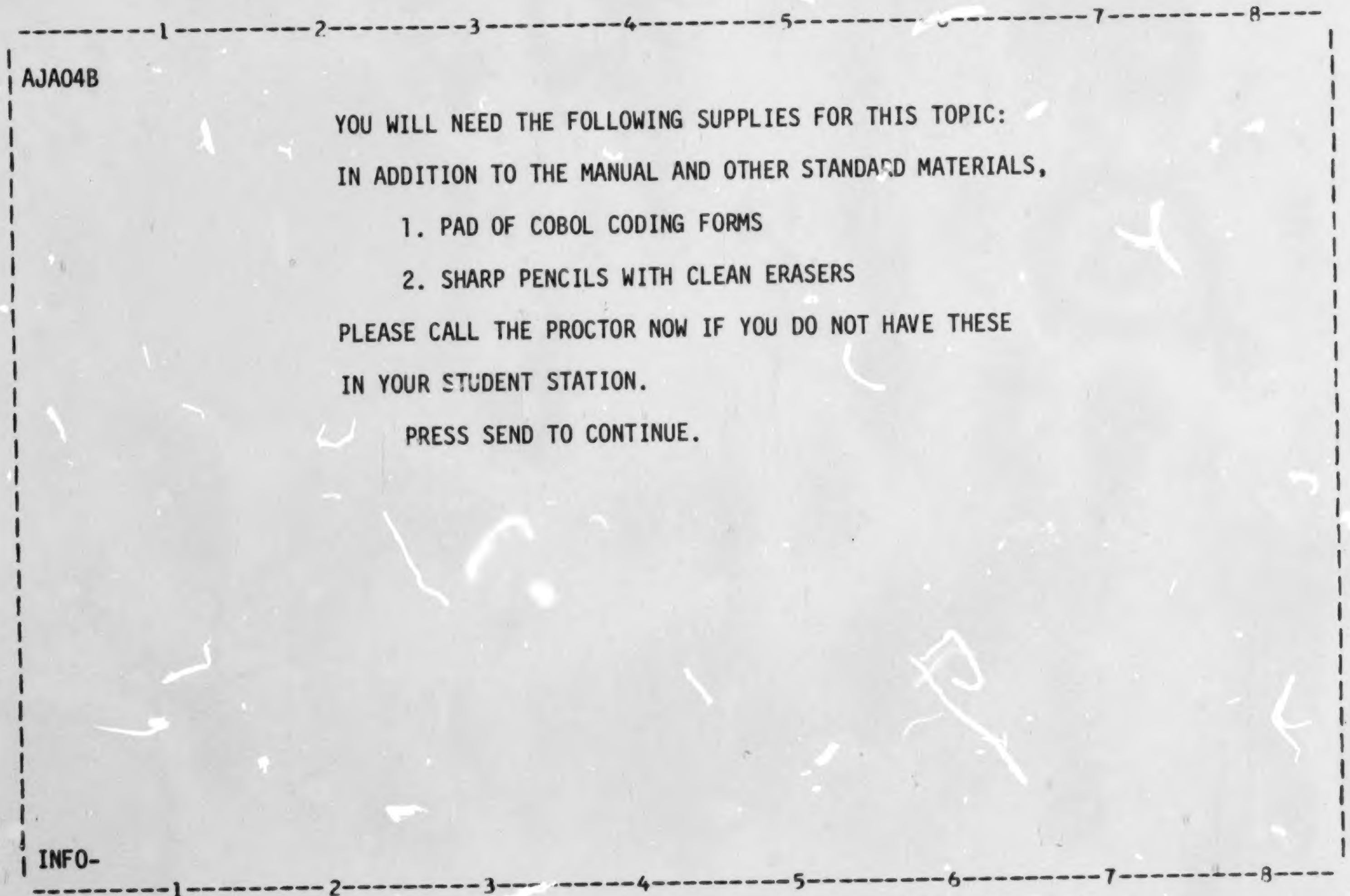
SAMPLE 11. A-Section: Standard Page 02 for Descriptions

```
---------1---------2---------3---------4---------5---------6---------7---------8----
                                       OBJECTIVES

         GIVEN:

                   1. NARRATIVE SPECIFICATIONS FOR DIVISION A CRITERION PROBLEM (CODE A
                      SIMPLE PROCEDURE DIVISION)

                   2. DATA-NAMES EXCEPTING PROCEDURE-NAMES

                   3. ACCESS TO REFERENCES CONTAINING FORMATS, RESERVED WORDS, RULES FOR CODING
                      ON A FORM. AND A CHECKLIST

                   4. A SAMPLE OF COMPLETE CODING FOR A SIMILAR PROCEDURE DIVISION ON A
                      CODING SHEET.

         THEN:     THE STUDENT WILL

                   1. CODE A COMPLETE PROCEDURE DIVISION ON A COBOL FORM

                   2. ANSWER QUESTIONS ABOUT HIS ENTRIES INTO NAME, IDENTIFICATION AND
                      SEQUENCE FIELDS

                   3. ANSWER QUESTIONS ABOUT HIS CODING AND THE AREAS (A AND B) IN WHICH VARIOUS
                      LINES APPEAR.

         CONSTRAINTS:

                   1. STUDENT MUST CODE OFF-LINE ONTO CODING FORM.




INFO-
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 12. A-SECTION: STANDARD PAGE 03 FOR OBJECTIVES

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|AJA04A                                                                             |
|                                                                                   |
|                                   DIRECTIONS                                      |
|                                                                                   |
|          REMEMBER, THE INFO OPTION IS AVAILABLE TO YOU AS YOU RECEIVE             |
|          INSTRUCTION IN THE COURSE. THIS OPTION IS CLOSED DURING THE              |
|          QUIZ. PRESS SEND.                                                        |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 13. A-SECTION: STANDARD PAGE Ø4 FOR DIRECTIONS



```
----------1---------2---------3---------4---------5---------6---------7---------8----
| AJA04B                                                                              |
|                    YOU WILL NEED THE FOLLOWING SUPPLIES FOR THIS TOPIC:              |
|                    IN ADDITION TO THE MANUAL AND OTHER STANDARD MATERIALS,           |
|                        1. PAD OF COBOL CODING FORMS                                  |
|                        2. SHARP PENCILS WITH CLEAN ERASERS                           |
|                    PLEASE CALL THE PROCTOR NOW IF YOU DO NOT HAVE THESE              |
|                    IN YOUR STUDENT STATION.                                          |
|                        PRESS SEND TO CONTINUE.                                       |
|                                                                                      |
| INFO-                                                                               |
----------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 14. A-SECTION: STANDARD PAGE 04 FOR MATERIALS (FOR STUDENTS)

```
---------1---------2---------3---------4---------5---------6---------7---------8----
| AJA05A
|                                  PROCTOR GUIDE
|        AJA04B - IF THE STUDENT DOES NOT HAVE COBOL CODING FORMS, SHARP PENCILS,
|                 ETC., PROCTOR SHOULD PROVIDE THEM.
|        AJ130A - COLLECT STUDENT'S COMPLETED CODING SHEET AND:
|                   (1) IF DAY SHIFT, WRITE ON IT "STUDENT JOB." "CALL
|                       273 WHEN DONE." SNED TO THE COMPUTER CENTER FOR
|                       KEYPUNCHING.
|                   (2) IF NIGHT SHIFT, HAVE A PROCTOR KEYPUNCH THE PROGRAM.
|                 IN EITHER CASE, HAVE THE STUDENT GO ON TO THE NEXT TOPIC.
|                 WHEN CARDS ARE RETURNED FROM THE COMPUTER CENTER, PUT THEM
|                 IN THE PROPER PLACE IN THE PROGRAM A MASTER DECK AND SEND
|                 THE JOB TO THE COMPUTER CENTER WITH INSTRUCTIONS TO CALL AS
|                 SOON AS IT IS READY.
|
| INFO-
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 15. A-SECTION: STANDARD PAGE 05 FOR PROCTOR GUIDE

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|AJA06A                                                                             |
|                                   MATERIALS                                       |
|                                                                                   |
|              STUDENT REFERENCE MANUAL.                                             |
|                                                                                   |
|              P-SCOPES AJA00P      ON AT AJA00A        OFF AT AJA01A                |
|                                                                                   |
|                       AJA10P           AJA12A               AJA14A                |
|                                                                                   |
|                       AJA15P           AJA14A               AJA16A                |
|                                                                                   |
|                       AJA55P           AJA16A               AJA18A                |
|                                                                                   |
|                       AJA20P           AJA18A               AJA20A                |
|                                                                                   |
|                       AJA25P           AJA20A               AJA22A                |
|                                                                                   |
|                       AJA30P           AJA22F               AJA22G                |
|                                                                                   |
|                       AJA35P           AJA22G               AJA22H                |
|                                                                                   |
|                       AJA40P           AJA22H               AJA24A                |
|                                                                                   |
|                       AJA45P           AJA24A               AJA28A                |
|                                                                                   |
|                       AJA50P           AJA28A               AJA30A                |
|                                                                                   |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 16. A-SECTION: STANDARD PAGE 06 FOR MATERIALS (FOR PROCTORS)

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|                          RESPONSE ANALYSIS                                        |
|                                                                                   |
|          SECTION T CONTAINS ONLY PRESS SEND RESPONSES (PSR).                      |
|                                                                                   |
|          SECTION P CONTAINS 15 COBOL CODING RESPONSES (CCR).                      |
|                                                                                   |
|              TRY 1. IF CCR IS CORRECT, GO TO CAF AND NEXT A.                      |
|                                                                                   |
|                     ELSE, GO TO TAF AND DAF.                                      |
|                                                                                   |
|              TRY 2. SAME AS FOR TRY 1. TAF IS ZAP.                                |
|                                                                                   |
|              TRY 3. IF CCR IS CORRECT, GO TO CAF AND NEXT A.                      |
|                                                                                   |
|                     ELSE, GO TO PROCTOR REMINDER ROUTINE.                         |
|                                                                                   |
|              NOTE:  STUDENT MAKES UP DATA-NAMES FOR THESE ANSWERS:                |
|                                                                                   |
|                     BGA30A         BGA40A         BGA52A                          |
|                     BGA34A         BGA44A                                         |
|                     BGA38A         BGA48A                                         |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 17. A-SECTION: STANDARD PAGE 07 FOR RESPONSE ANALYSIS

```
---------1---------2---------3---------4---------5---------6---------7---------8----
 COBOL CODING RESPONSE (CCR) WITH ZAP-DAF ROUTINE.
 HEADER                    353   354   383   357   362
 FD ASSIGNED-NAME          353   354   383   376   362
 FD ELECTIVE-NAME-2 +      372   360   383   376   362
 LR CLAUSE                 353   354   383         362
 DR ASSIGNED-NAME          353   354   383   376   362
 DR ELECTIVE-NAME-2 +      372   360   383   376   362
 RC CLAUSE                 353   354   383   377   362
 01 + PIC CLAUSE           372   370   373   376   362
 77 ASSIGNED-NAME          353   354   370   376   362
 77 ELECTIVE-NAME-2 +      372   370   360   376   362
 77 + PIC CLAUSE           372   370   373   376   362
 USAGE-VALUE               353   354   380   381   362
 PIC-USAGE-VALUE           372   373   380   381   362
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 18. A-SECTION: SUPPLEMENT TO 07 FOR RESPONSE ANALYSIS

```
---------1---------2---------3---------4---------5---------6---------7---------8----

                              KEY TERMS

             COBOL CODING FORM             IDENTIFICATION FIELD

             CODING FORM                   JOB CARD

             COLUMNS 8 - 11                PARAGRAPH-NAME

             COLUMNS 8 - 72                PROCEDURE-NAME

             COLUMNS 73 - 80               PROGRAM-NAME

             DIVISION-HEADER               REPLACEMENT TECHNIQUE

             DIVISION-NAME                 ROUTINE

             FORMAT                        SERIAL NUMBER

                                           TURNAROUND TIME

INFO-
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 19. A-SECTION: STANDARD PAGE 08 FOR KEY TERMS

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|                                                                                   |
|                             PROGRAMMER-SUPPLIED-NAMES                             |
|                                                                                   |
|             CARD-FILE                                                             |
|                                                                                   |
|             HOUSEKEEPING                                                          |
|                                                                                   |
|             LIST-FILE                                                             |
|                                                                                   |
|             PROCESS-DATA                                                          |
|                                                                                   |
|             RECORD-IN                                                             |
|                                                                                   |
|             RECORD-OUT                                                            |
|                                                                                   |
|             WRAP-UP                                                               |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 20. A-SECTION: STANDARD PAGE 09 FOR SPECIAL TERMS

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|                                                                                   |
|                 THE PROGRAMMER-NAMES (QUIZ)                                        |
|                                                                                   |
|           THERE ARE NO PROGRAMMER-SUPPLIED-NAMES IN THIS QUIZ. STUDENT USES        |
|                                                                                   |
|           PSN'S FROM A-MASTER PROGRAM IN HIS CODING.                               |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 21. A-SECTION: STANDARD PAGE 09 FOR SPECIAL TERMS (QUIZ)

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|BGA10A                                                                             |
|                                                                                   |
|        THIS TOPIC HELPS YOU PUT TOGETHER A COMPLETE COBOL PROGRAM.                |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 22. T-SECTION: STANDARD PAGE 10 FOR TRANSITION

```
001810     BFA44A    1    P    P02    X    S    BFA44G+     12354    CCP
F01406     BFA44F    0    P    RAF    0    0                00051    PSR
F01407     BFA44G    1    P    P03    0    S    BFA44H-M    00051    ASR
F01508     BFA44H    1    P    P04    0    S    BFA44J      00051    PSR
F01409     BFA44I    1    P    P05    0    S    BFA44J      00051    PSR
F01410     BFA44J    1    P    P06    0    S    BFA44A      00051    PSR
F01411     BFA44K    1    P    P07    0    S    BFA44L      00051    PSR
F01412     BFA44L    1    P    P08    0    S    BFA44N      00041    PSR
F01413     BFA44M    1    P    P09    0    S    BFA44N+     00051    PSR
F01414     BFA44N    1    P    P10    0    S    BFA44M+     00061    PSR
F01415     BFA44O    1    P    P12    0    S    BFA44J      00051    PSR
F01416     BFA44Q    1    P    P13    0    S    BFA44J      00051    PSR
F01417     BFA44R    1    P    P14    0    S    BFA44J      00051    PSR
F01418     BFA44S    1    P    P15    0    S    BFA44T      00051    PSR
F01419     BFA44T    1    P    P16    0    S    BFA44U      00041    PSR
F01420     BFA44U    1    P    P17    0    S    BFA44M+     00061    PSR
F01421     BFA44V    1    P    P19    0    S    BFA44J      00051    PSR
F01422     BFA44W    1    P    P20    0    S    BFA44J      00051    PSR
F01423     BFA44X    1    P    P21    0    S    BFA44J      00051    PSR
F01424     BFA44Y    1    P    P22    0    S    BFA44Z      00051    PSR
F01425     BFA44Z    1    P    P23    0    S    BFA46A      00041    PSR
```

SAMPLE 23. TITLE LISTING OF ALL ELEMENTS ON A CRT PAGE

```
---------1---------2---------3---------4---------5---------6---------7---------8----

DIV: B   MOD:  A   VER:  1   OBJ:  1   DATE:  9/22/71   INIT:FFK

OBJ DES:   CODE AN IDENTIFICATION DIVISION

SCORABLE EVENTS:

A.  CORRECT A-B MARGIN          D.  VALID P.S.N.        G.

B.  CORRECT PUNCTUATION         E.  CORRECT LINE NO.    H.

C.  CORRECT CRW                 F.  CORRECT SENTENCE    I.

EVENTS       A   B   C   D   E   F   G   H   I   MAX   Q-SCORE    IDM

POINTS       1   1   1   3   2   1               PTS      0        0
                                                         1-5       1
C   BA110A   1   1   2       1   1                7     6-10       2
    BA110B   1   2   1   1   1                    9    11-14       3
W   BA110C   1   1   1   1                        6    15-17       4
    BA110D   1   1   1  -3 +                      3    18-20       5
I                                                        21        6
                                                         22        7
P                                                        23        8
                                                         24        9
S                                                        25       10

TOTALS       4   5   5   3   2   1               25

OTHER INSTR:   SEE BA1018

---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 24. Q-SECTION: STANDARD PAGE 101 FOP QUIZ SCORE COMPUTATIONS

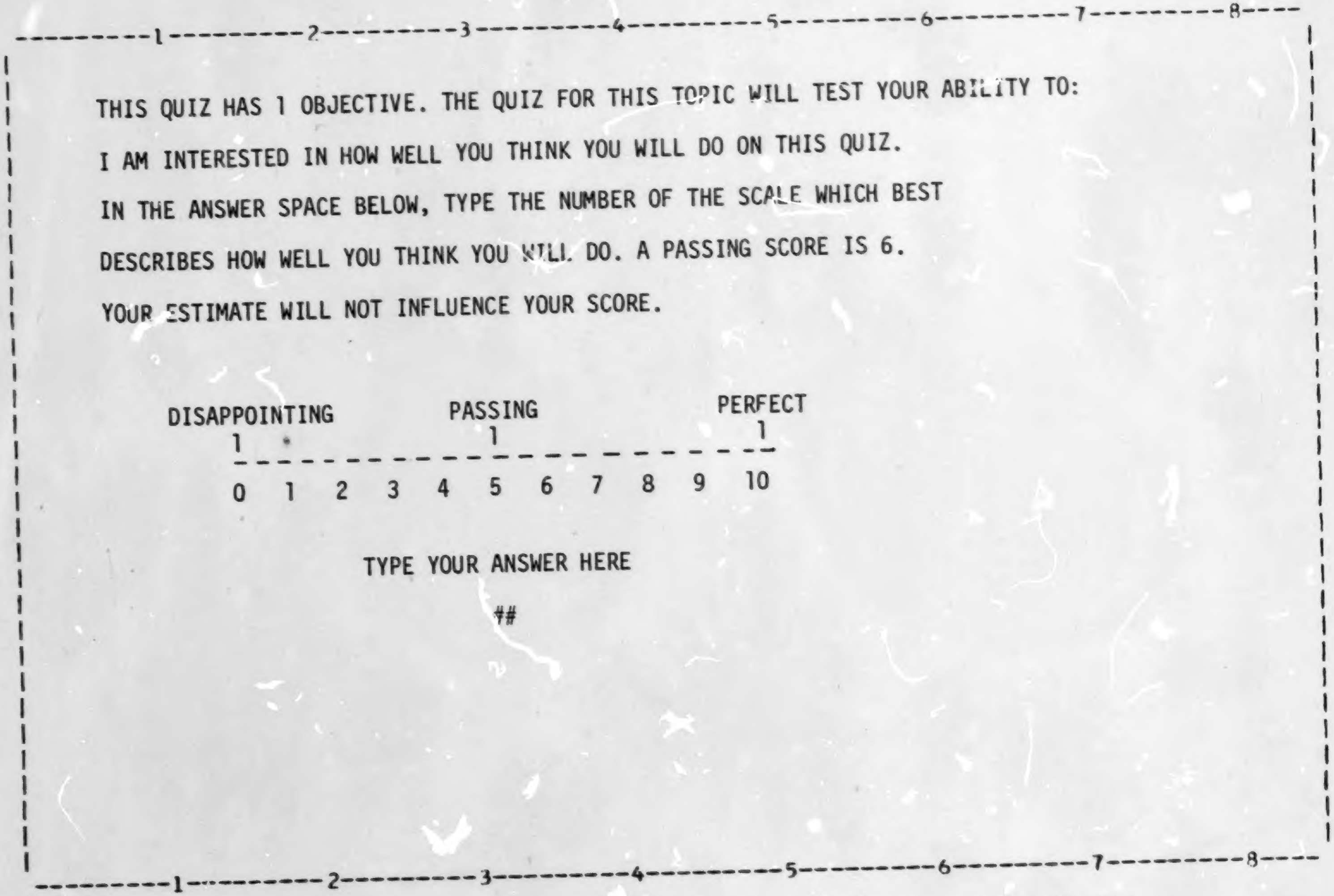
```
---------1---------2---------3---------4---------5---------6---------7---------8----
|BA101B                                                                            |
|                                                                                  |
|                         COMPUTATION OF Q-SECTION SCORE                           |
|                                                                                  |
|                         (CONTINUED FROM BA101A, INCLUDES                         |
|                                                                                  |
|                          OTHER INSTRUCTIONS)                                     |
|                                                                                  |
|       OTHER INSTRUCTIONS:                                                        |
|                                                                                  |
|           MAA FOR CORRECT LINE NUMBER:                                           |
|                BA110A 01   ID&                                                   |
|                BA110B 02   PR*G&ID&                                              |
|                                                                                  |
|           LINES 3 (BA110C) AND 4 (BA110D) MAY BE IN  REVERSED ORDER.             |
|                                                                                  |
|           MAA LINE 03 (BA110C):                                                  |
|                O* A*T&R                                                          |
|                ANYTHING AFTER THE SPACE FOLLOWING "AUTHOR" IS A VALID            |
|                    PROGRAMMER-SUPPLIED NAME.                                     |
|                                                                                  |
|           MAA LINE 04 (BA110D):                                                  |
|                O* D* T&-C&                                                       |
|                +  SUBTRACT 3 FOR ATTEMPT TO SUPPLY ANYTHING FOLLOWING:           |
|                    DATE-COMPILED.                                                |
|                                                                                  |
|           LINE 02 (BA110B):  USE FUNCTION SYNTAX TO ANALYZE PROGRAM NAME.        |
|                                                                                  |
|                                                                                  |
|                                                                                  |
|                                                                                  |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

WHERE: MAA = MINIMUM ACCEPTABLE ANSWER
* = ANY CHARACTER
& = ANY CHARACTER OR STRING OF CHARACTERS

SAMPLE 25. Q-SECTION: SUPPLEMENT TO 101 FOR QUIZ SCORE COMPUTATIONS

```
---------1---------2---------3---------4---------5---------6---------7---------8----
BA102A

            THIS IS YOUR LAST CHANCE FOR THE INFO OPTION BEFORE THE QUIZ.
            THE QUESTIONS THAT FOLLOW COVER BOTH THE IDENTIFICATION AND
            ENVIRONMENT DIVISIONS. JUST PRESS SEND TO GO ON.

I[illegible]O-
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 26. Q-SECTION: STANDARD PAGE 1Ø2 FOR TRANSITION



THIS QUIZ HAS 1 OBJECTIVE. THE QUIZ FOR THIS TOPIC WILL TEST YOUR ABILITY TO:

I AM INTERESTED IN HOW WELL YOU THINK YOU WILL DO ON THIS QUIZ.

IN THE ANSWER SPACE BELOW, TYPE THE NUMBER OF THE SCALE WHICH BEST

DESCRIBES HOW WELL YOU THINK YOU WILL DO. A PASSING SCORE IS 6.

YOUR ESTIMATE WILL NOT INFLUENCE YOUR SCORE.

```
DISAPPOINTING         PASSING          PERFECT
  1                     1                 1
  - - - - - - - - - - - - - - - - - - - - -
  0   1   2   3   4   5   6   7   8   9   10
```

TYPE YOUR ANSWER HERE

##

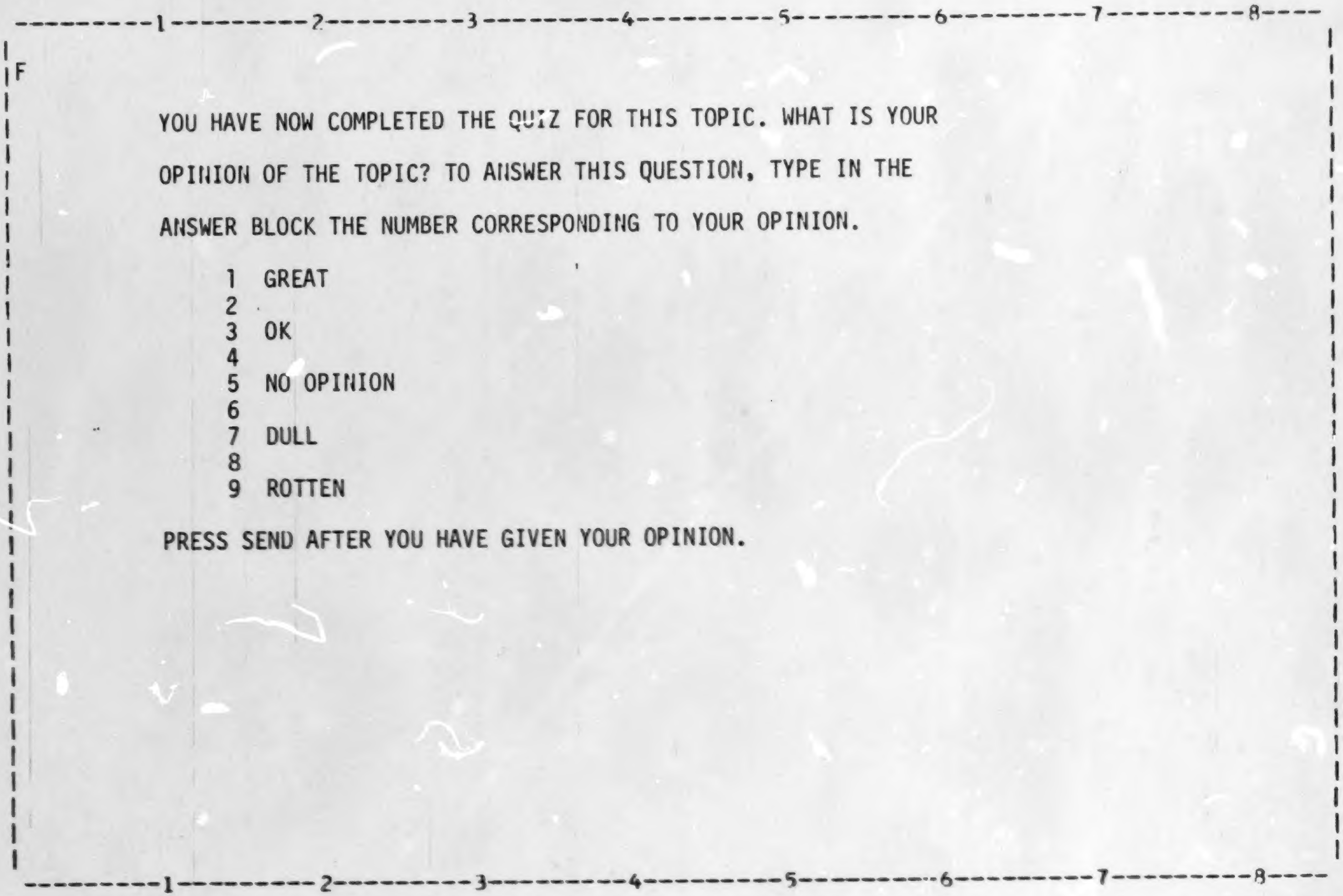
SAMPLE 27. Q-SECTION: STANDARD PAGE 103 FOR LEVEL OF ASPIRATION (LOA) TESTING

```
---------1---------2---------3---------4---------5---------6---------7---------8----
BA104A
       THE COMPLETE QUIZ FOR THIS TOPIC CONSISTS OF 4 QUESTIONS.

            1. WRITE AN IDENTIFICATION DIVISION ON THE CRT AND THE CODING FORM.

            2. WRITE AN ENVIRONMENT DIVISION IN THE SAME WAYS.

            3. ANSWER A QUESTION ABOUT THE ENVIRONMENT DIVISION.

            4. ANSWER ANOTHER QUESTION ABOUT YOUR CODING.

       CAUTION

                READ ALL DIRECTIONS CAREFULLY.

                DOUBLE-CHECK ALL ANSWERS BEFORE PRESSING SEND.

       REMEMBER

                INFO IS CLOSED DURING THE QUIZ.

                COMMENTS ARE WELCOMED AT THE END.

       IF YOU HAVE TO REPEAT THE QUIZ, YOU WILL BE ABLE TO SKIP ANY QUESTION

       WITH A COMPLETELY CORRECT ANSWER ON THE PREVIOUS TRY.

---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 28. Q-SECTION: STANDARD PAGE 104 FOR QUIZ DIRECTIONS

```
---------1---------2---------3---------4---------5---------6---------7---------8----
                                   RESPONSE ANALYSIS
       PROGRAMMING SEQUENCE FOR TOPIC BA QUIZ ON TRY 2 AND TRY XX.
           GO TO 102A AND 104A FOR ALL STUDENTS.
           IF 110A = WA, GO TO 108A, 110A, 120A.
           IF 116A = WA, GO TO 114A, 116A, 120A.
           IF 110A AND 116A = WA, GO TO 108A, 110A, 114A. 116A, 120.
           IF 122A = WA, GO TO 122A.
           IF 124A = WA, GO TO 124A.
           THEN, GO TO CATPER AND APPROPRIATE ROLL-ONS. *
           LAST, GO TO 126A.
           STUDENTS WHO PASS QUIZ ON TRY 1 OR 2 WITHOUT PERFECT SCORES ALSO
           RECEIVE CATQR1 (D 3417) OR CAYQR2 (D 3416) AND QUESTIONS WITH
           CORRECT ANSWERS ROLLED INTO ANSWER SPACES OR QRA'S.
               FOR THIS TOPIC, SEE BA1QRA (D 3422).
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 29. Q-SECTION: STANDARD PAGE 106 FOR QUIZ SEQUENCE

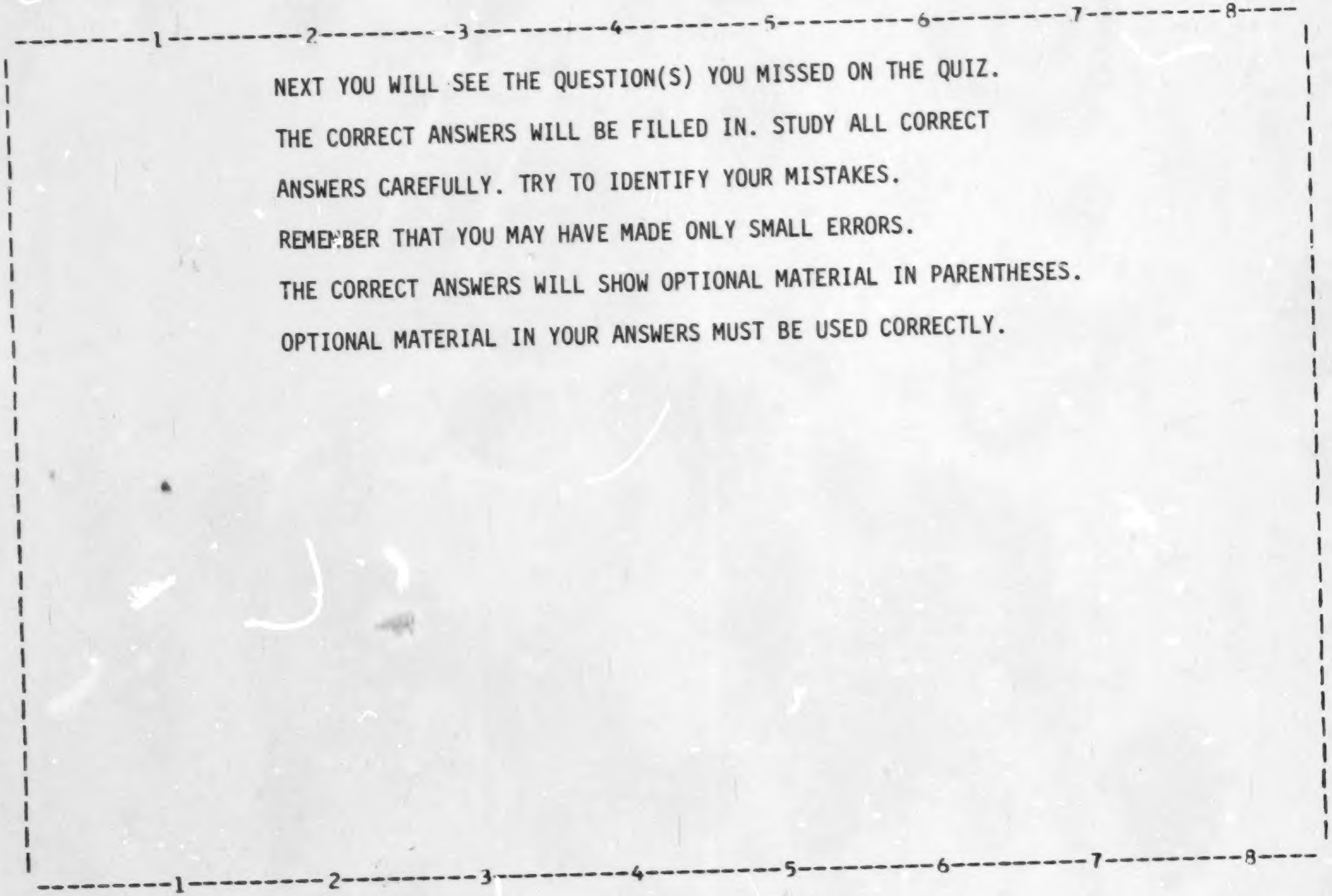


```
F

YOU HAVE NOW COMPLETED THE QUIZ FOR THIS TOPIC. WHAT IS YOUR

OPINION OF THE TOPIC? TO ANSWER THIS QUESTION, TYPE IN THE

ANSWER BLOCK THE NUMBER CORRESPONDING TO YOUR OPINION.

    1  GREAT
    2
    3  OK
    4
    5  NO OPINION
    6
    7  DULL
    8
    9  ROTTEN

PRESS SEND AFTER YOU HAVE GIVEN YOUR OPINION.
```

SAMPLE 30. Q-SECTION: STANDARD PAGE FOR STUDENT OPINIONS

```
---------1---------2---------3---------4---------5---------6---------7---------8----

         THE PREVIOUS QUIZ TESTED YOUR ABILITY TO


                                                 NUMBER SCORE PERFORMANCE


         BEFORE THE QUIZ YOUR ESTIMATE WAS:

         YOU      THIS OBJECTIVE WITH:

          YOUR ANSWERS SHOW THE FOLLOWING TYPES OF ERRORS:




                     SEQUENCE OF STATEMENTS NOT IN ORDER

                     INVALID PROGRAMMER-SUPPLIED-NAME(S)

                     INCORRECT PUNCTUATION

                     INCORRECT MARGIN(S)

                     INVALID COBOL RESERVED WORD


---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 31. Q-SECTION: STANDARD PAGE FOR PERFORMANCE EVALUATIONS



NEXT YOU WILL SEE THE QUESTION(S) YOU MISSED ON THE QUIZ.

THE CORRECT ANSWERS WILL BE FILLED IN. STUDY ALL CORRECT

ANSWERS CAREFULLY. TRY TO IDENTIFY YOUR MISTAKES.

REMEMBER THAT YOU MAY HAVE MADE ONLY SMALL ERRORS.

THE CORRECT ANSWERS WILL SHOW OPTIONAL MATERIAL IN PARENTHESES.

OPTIONAL MATERIAL IN YOUR ANSWERS MUST BE USED CORRECTLY.

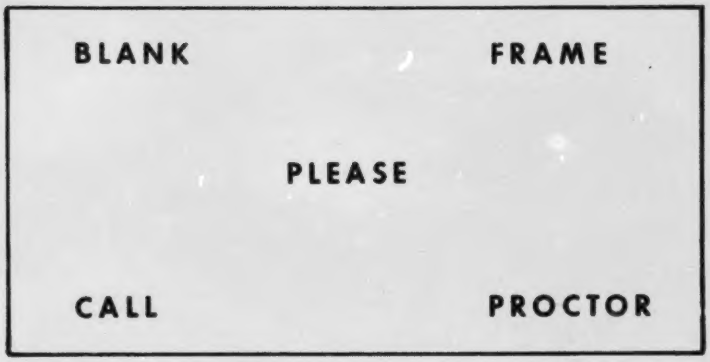
SAMPLE 32. Q-Section: Standard Page for Quiz Review

```
---------1---------2---------3---------4---------5---------6---------7---------8----
        MARGIN  A   H
                1   1

          01 ENVIRONMENT DIVISION.

          02 CONFIGURATION SECTION.

          03 SOURCE-COMPUTER.  IBM-360-40.

          04 OBJECT-COMPUTER.  IBM-360-40.

          05 INPUT-OUTPUT SECTION.

          06 FILE-CONTROL.

          07      SELECT DECK, ASSIGN TO UT-S-READER.

          08      SELECT LISTING, ASSIGN TO UT-S-PRINTER.

---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE. 33. Master Element Showing Correct Answers for Quiz Review

```
---------1---------2---------3---------4---------5---------6---------7---------8----
|BA126A                                                                             |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                  THE QUIZ IS OVER.                                                |
|                                                                                   |
|                     IF YOU WISH, COMMENT BELOW IN THE USUAL MANNER.               |
|                                                                                   |
|                        PRESS SEND TO CONTINUE.                                    |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|                                                                                   |
|INFO-                                                                              |
---------1---------2---------3---------4---------5---------6---------7---------8----
```

SAMPLE 34. Q-Section: Standard Page for End of Quiz



BLANK **FRAME**

PLEASE

CALL **PROCTOR**

SAMPLE 35. Blank Frame Used to Separate Groups of Visuals

Appendix F

MNEMONICS AND MESSAGES

KEY TO TITLE PURPOSE FIELD

AAF = ANY-ANSWER-FEEDBACK
AA9 = ALTERNATE-ANSWER-FBK-N
ANA = ANALYSIS
APP = APPLICATION
ASD = ANIMATION-SEQUENCE-DISPLAY
BET = BET-FOR-ERROR-GAME
BF9 = BETTING-FEEDBACK-N
BGF = BEGIN-GAME-FEEDBACK
BRN = BRANCH
CAF = CORRECT-ANSWER-FEEDBACK
CA9 = CORRECT-ANSWER-FEEDBACK-N
CME = CONFIDENCE-MASTER-ELEMENT
COM = COMPUTATION
COP = COPY-PRACTICE
DAF = DIAGNOSTIC-ANALYSIS-FEEDBACK
DEB = DEBUG
DEF = DEFINITION
DES = DESCRIPTION
DIR = DIRECTIONS
DOC = DOCUMENTATION
EMF = ERROR-MESSAGE-FEEDBACK
END = END-OF-SECTION
EOP = END-OF-PRACTICE
EOQ = END-OF-QUIZ
ERM = ERROR-MESSAGE
ERR = ERROR
EVA = EVALUATION
EXP = EXPLANATION
FAA = FEEDBACK-ANSWER-A ETC
FYN = FEEDBACK-YES-NO-RESPONSE
HAT = HALF-TITLE
INS = INSERT
INT = INTRODUCTION
JOG = JOG-STUDENT
KEY = KEY-TERMS
LIT = LITERAL
LOA = LEVEL-OF-ASPIRATION
LOO = LOOP

MAT = MATERIALS
MEN = MENU
ME9 = MASTER-ELEMENT-N
OBJ = OBJECTIVES
OPI = OPINION
PER = PERFORMANCE
PRC = PROCTOR
PRE = PREREQUISITES
PRG = PROCTOR-GUIDE
PR9 = PROBLEM-N
PSN = PROGRAMMER-SUPPLIED-NAMES
QU9 = QUIZ-QUESTION-N
Q99 = QUIZ-QUESTION-N
RAD = RESPONSE-ANALYSIS-DISPLAY
RAF = RESPONSE-ANALYSIS-FEEDBACK
REC = RECOGNITION
REF = REFERENCE
RES = RESET
REV = REVIEW
ROQ = ROLL-ON-QUESTION
ROT = ROLL-ON-TEXT
RT9 = ROLL-ON-TEXT-N
SEQ = SEQUENCE
SE9 = SEQUENCE-N
SOL = SOLUTION
SPC = SPECIFICATIONS
SUM = SUMMARY
TAA = TRY-AGAIN-ANSWER-A ETC
TAD = TRY-AGAIN-DISPLAY
TAF = TRY-AGAIN-FEEDBACK
TA9 = TRY-AGAIN-FEEDBACK-N
TRN = TRANSITION
TRY = TRY-OPTIONAL-PRACTICE
UND = UNDERSTANDING
WAA = WRONG-ANSWER-A ETC
WAF = WRONG-ANSWER-FEEDBACK
WAQ = WRONG-ANSWER-QUIZ-OPTION
WAR = WRONG-ANSWER-REVIEW-OPTION
WA9 = WRONG-ANSWER-FEEDBACK-N

KEY TO TITLE RESPONSE FIELD

ASR = AUTO-SYSTEM-RESPONSE
BN9 = BOTH-NEITHER-RESPONSE
CCI = CODING-RESPONSE,9-ENTRIES
CCJ = CODING-RESPONSE,10-ENTRIES
CCK = CODING-RESPONSE,11-ENTRIES
CCQ = CODING-RESPONSE,19-ENTRIES
CCR = COBOL-CODING-RESPONSE
CFR = CODING-FORMAT-RESPONSE
CMR = COMMENT-RESPONSE-EXPECTED
C99 = CODING-RESPONSE,99-ENTRIES
DER = DELETION-RESPONSE
DOT = RESET-RESPONSE
EGR = ERROR-GAME-RESPONSE
FC9 = FORCED-CHOICE-N-CHOICES
FIR = FILL-IN-RESPONSE
GLR = GLOSSARY-REQUEST
INF = RESET-INFO-BLOCK-RESPONSE
INR = INSERTION-RESPONSE
LPB = LIGHT-PEN-RESP-2-CHOICES
LPC = LIGHT-PEN-RESP-3-CHOICES
LPR = LIGHT-PEN-RESPONSE
LP9 = LIGHT-PEN-RESPONSE-N-CHOICES
MA9 = MATCH-RESPONSE-N-CHOICES
MC9 = MULTIPLE-CHOICE-N-CHOICES
OLR = OFF-LINE-RESPONSE
OPR = OPTION-RESPONSE-EXPECTED
PRC = PROCTOR-CALL
PSR = PRESS-SEND-RESPONSE
RER = REPLACEMENT-RESPONSE
RSR = RATING-SCALE-RESPONSE
SER = SEQUENCE-RESPONSE
SE9 = SEQUENCE-N-RESPONSE
SIR = SPECIAL-INFO-REQUEST
SFR = SIGN-OFF-RESPONSE
TFR = TRUE-FALSE-RESPONSE
WID = WRITE-IN-RESPONSE-4-CHOICE
WIH = WRITE-IN-RESPONSE-6-CHOICE
WIR = WRITE-IN-RESPONSE
WIT = WRITE-IN-RESPONSE-TRY-AGAIN
YNR = YES-NO-RESPONSE

DIAGNOSTIC ANALYSIS FEEDBACKS (DAFs) USED IN COBOL2

Disc Location	Label	Text
F00225	ZF0225	CHECK YOUR LINE NUMBER.
F00353	ZF0353	CHECK YOUR MARGINS.
F00354	ZF0354	CHECK YOUR PUNCTUATION.
F00355	ZF0355	CHECK YOUR SPELLING.
F00356	ZF0356	CHECK FOR ILLEGAL CHARACTERS.
F00357	ZF0357	CHECK FOR ILLEGAL SPACES.
F00358	ZF0358	CHECK FOR ILLEGAL USE OF COBOL RESERVED WORDS.
F00360	ZF0360	ALL NAMES MUST BE DIFFERENT OR UNIQUE.
F00361	ZF0361	EACH NAME MUST CONTAIN AT LEAST 1 LETTER.
F00362	ZF0362	CHECK THE FORMAT IN YOUR MANUAL.
F00365	ZF0365	CHECK YOUR DATA-NAME.
F00366	ZF0366	CHECK YOUR COMMENT-ENTRY.
F00367	ZF0367	CHECK YOUR COMPUTER-NAME.
F00368	ZF0368	CHECK YOUR COUNTER INCREMENT NUMBER.
F00369	ZF0369	CHECK YOUR DATA-NAME QUALIFICATION.
F00370	ZF0370	CHECK YOUR LEVEL NUMBER.
F00371	ZF0371	CHECK YOUR LITERAL.
F00372	ZF0372	CHECK YOUR MARGINS AND/OR PUNCTUATION.
F00373	ZF0373	CHECK YOUR PICTURE CLAUSE.
F00374	ZF0374	CHECK YOUR PROCEDURE-NAME.
F00375	ZF0375	CHECK YOUR PROGRAM-NAME.
F00376	ZF0376	CHECK YOUR PROGRAMMER-SUPPLIED-NAME.
F00377	ZF0377	CHECK YOUR RECORD LENGTH.
F00378	ZF0378	CHECK YOUR SEQUENCE NUMBER.
F00379	ZF0379	CHECK YOUR SYSTEM-NAME.
F00380	ZF0380	CHECK YOUR USAGE CLAUSE.
F00381	ZF0381	CHECK YOUR VALUE CLAUSE.
F00382	ZF0382	ENTER PARAGRAPH-HEADER ONLY. COMPUTER SUPPLIES DATE.
F00383	ZF0383	CHECK THE RESERVED WORDS.
F00384	ZF0384	ALL DATA-NAMES MUST BE DIFFERENT.
F00385	ZF0385	IS THERE A PERIOD RIGHT AFTER YOUR PROCEDURE-NAME?
F00386	ZF0386	CHECK THE RULES IN YOUR MANUAL.
F00387	ZF0387	PROCEDURE-NAMES SHOULD NEITHER BEGIN NOR END WITH A HYPHEN (-).
F00388	ZF0388	PROCEDURE-NAMES SHOULD NOT BE LONGER THAN 30 CHARACTERS.
F01788	ZF1788	DID YOU BEGIN IN THE CORRECT MARGIN?
F01789	ZF1789	DID YOU USE CORRECT PUNCTUATION?
F01790	ZF1790	DID YOU USE CORRECT DATA-NAMES?
F01791	ZF1791	DID YOU USE COBOL WORDS CORRECTLY?
F01792	ZF1792	DID YOU FOLLOW THE CORRECT FORMAT EXACTLY?

STANDARD FEEDBACKS (ZAPs) USED IN COBOL2

Disc Location	Label	Text
F00351	ZF0351	PLEASE ANSWER AS DIRECTED.
F00352	ZF0352	YOU SEEM TO BE HAVING A PROBLEM WITH THIS QUESTION. MAYBE THE PROCTOR CAN HELP YOU. PRESS THE PANIC BUTTON FOR ASSISTANCE OR PRESS SEND TO TRY AGAIN ON YOUR OWN.
F00359	ZF0359	PLEASE TYPE IN REQUIRED RESPONSE.
F00363	ZF0363	PLEASE CALL THE PROCTOR.
F00364	ZF0364	PRESS SEND AND TRY AGAIN.
F01825	ZF1825	IF YOU SEND ANOTHER BLANK LINE, I'LL GIVE YOU SOME HINTS. PRESS SEND AND TRY AGAIN.
F02316	ZF2316	DID YOU SEND EVERYTHING YOU WANTED? TYPE YES OR NO HERE AND PRESS SEND.
F02317	ZF2317	TRY AGAIN. REPOSITION THE CURSOR AND SEND YOUR ANSWER(S).

Appendix G

LISTING OF GLOSSARY TERMS AND DIAGNOSTICS

A
ABBREVIATION
ACRONYM
ADD
ADD 1
ALPHA
ALPHANUMERIC
ALPHANUMERIC KEY
A-MARGIN
ARE
AREA
AREA-NAME
ASSIGN
ASSIGN TO
AT
AT END
AT END CLAUSE
AT SYMBOL
AUTHOR
B
BETTING
BIT
BLANK
BLOCK
B-MARGIN
BRANCH
BRANCHES
BY
C
CAI
CALL
CARD-READER
CARRIAGE KEY
CASCADING ERROR
CATHODE
CATHODE RAY TUBE
CBL
C-BUTTON
CC
CHARACTER
CHARACTERS
CIRCLE
CLASS
CLAUSE
CLEAR
CLOSE
COBOL
COBOL CODING FORM
COBOL ENTRY
COBOL FORMAT
COBOL LANGUAGE ELEMENT
COBOL PUNCTUATION
COBOL RESERVED WORD
COBOL RULE
COBOL SPELLING
COBOL STATEMENT
COBOL WORD
COBOL2
CODE
CODING
CODING FORM
COLUMN
COMPILER
COMMENT
COMPUTATIONAL
COMPUTE
COMPUTER
COMPUTER PROGRAMMING
CONDITION
CONDITIONAL
CONDITIONAL STATEMENT
CONDITION FALSE
CONDITION TRUE
CONFIGURATION
CONFIGURATION SECTION
CONNECTOR
CONNECTOR SYMBOL
CONSOLE
CORE
COUNT
COUNTER
CR KEY
CRT
CRT DISPLAY TERMINAL
CRW
CUBICLE

CURSOR
C-552
D
DA
DATA
DATA AREA
DATA CRW
DATA DESCRIPTION
DATA DIVISION
DATA ENTRY
DATE FIELD
DATA ITEM
DATA LEVEL
DATA-NAME
DATA-NAMES
DATA PROCESSING
DATA RECORD
DATA SIZE
DATE
DATE-COMPILED
DEBUG
DEBUGGING
DECISION
DECISION BLOCK
DECISION SYMBOL
DECK
DELETE
DEVICE
DIAGNOSTIC
DIAGNOSTICS
DIAMOND
DIGIT
DIGS
DISPLAY
DISPLAY STATEMENT
DIVIDE
DIVISION
DIVISION HEADER
DIVISION HEADERS
E
EDIT
ELEMENTARY ITEM
ELEMENTARY ITEMS
END
END OF FILE
ENTER
ENTRY
ENVIRONMENT
ENVIRONMENT DIVISION
EQUAL TO
ERROR
ERROR MESSAGE
ERROR MESSAGES
EXIT
EXTERNAL-NAME
FD
F
FEEDBACK
FILE
FILE-CONTROL
FILE CRW
FILE DESCRIPTION
FILE-NAME
FILE SECTION
FILLER
FLOW
FLOWCHART
FLOWCHART BLOCK
FLOWCHART SYMBOL
FORMAT
FORMATS
FORTRAN
FREE
G
GARBAGE
GENERATE
GLOSSARY
GO
GO TO
GREATER THAN
GROUP ITEM
H
HARDWARE
HEADER
HEADING
HELP
HOUSEKEEPING
HUMAN
HUMRRO
I
IBM
IBM-360-40
IDENTIFICATION
IDENTIFICATION CRW
IDENTIFICATION DIVISION
I ENJOY DATA PROCESSING
IF
IKF3001
IMPACT
INDEPENDENT WORK AREA
INDICATOR LAMP
INFO
INFO-OPTION
INFORMATION

INPUT
INPUT AREA
INPUT-CARD
INPUT CRW
INPUT FILE
INPUT-OUTPUT
INPUT-OUTPUT DEVICE
INPUT-OUTPUT SECTION
INPUT/OUTPUT SYMBOL
INPUT RECORD
INPUT SYMBOL
INSERT
INSTRUCTION
INTEGER
INTERMEDIATE DATA
I-O
IS
ITEM
ITEM DESCRIPTION
J
K
KEY
KEYPUNCH
L
LABEL RECORD
LABEL RECORD IS OMITTED
LAST
LEVEL NUMBER
LEVEL 77
LIGHT PEN
LIMIT
LINE
LINE-COUNTER
LINE-OUT
LIST
LISTING
LITERAL
LOCATION
LOCK
LOGIC
LOG OFF
LOOP
M
MARGIN
MEMORY
MNEMONIC
MODE
MODE KEY
MONIKER
MOVE
MOVE SPACES
N
NAME
NAMED
NAME-FIELD
NEXT SENTENCE
NO
NO-BRANCH
NON-NUMERIC LITERAL
NUMBER
NUMERIC
NUMERIC LITERAL
O
OBJECT-COMPUTER
OBJECTIVE
OF
OFF-LINE
OPEN
OPERAND
OPTION
OPTIONAL ENTRY
OUTPUT
OUTPUT AREA
OUTPUT CRW
OUTPUT FILE
OUTPUT RECORD
P
PAGE
PAGE-COUNTER
PARAGRAPH
PARAGRAPH HEADER
PARAGRAPH-NAME
PARALLELOGRAM
PERIOD
PICTURE
PICTURE CLAUSE
PLACE
POSITION
PRINTER
PRINTING OFF-LINE
PRINTOUT
PROBLEM
PROBLEM-SOLVING
PROCEDURE
PROCEDURE DIVISION
PROCEDURE-NAME
PROCEDURE-NAMES
PROCESS
PROCESSING
PROCESSING SYMBOL
PROCESSOR
PROCTOR

PROGRAM
PROGRAM ENTRY
PROGRAM INSTRUCTION
PROGRAMMER
PROGRAMMER-SUPPLIED-NAME
PROGRAM-NAME
PROGRAM STEP
PS
P-SCOPE
PUNCHED CARD
PUNCTUATION
Q
QUALIFICATION
QUALIFIER
QUOTE
R
RANDOM
RD
READ
READ INPUT FILE
RECORD
RECORD CRW
RECORD DESCRIPTION
RECORD FORMAT
RECORD-NAME
RECTANGLE
REMARKS
REMOTE ERROR
REPLACE
REPLACEMENT
REQUIRED ENTRY
RESERVED WORD
RESPONSE
RETURN
ROUTINE
RUN
S
SANDERS
SCREEN
SECTION
SECTION HEADER
SELECT
SEND
SEND BLOCK
SENTENCE
SEQUENCE
SEQUENCE NUMBER
SERIAL
SHIFT
SIGN
SIGN OFF
SIZE
SORT
SOURCE-COMPUTER
SOURCE PROGRAM
SPACE
SPACE KEY
SPACES
SPECIAL CHARACTER
SPECIFICATIONS
SPECS
SPELLING
START
STATEMENT
STEP
STOP
STOP RUN
STORAGE
STORAGE AREA
STUDENT
STUDENT AREA
STUDENT
STUDENT NUMBER
SUBTRACT
SYMBOL
SYNOPSIS
SYNTAX
SYSTEM
T
TAB
TEMPLATE
TERMINAL
TERMINAL SYMBOL
TERMINATE
TEST DATA
TO
TRANSFER
TUBE
TURNAROUND TIME
TYPE
U
UNIQUE
UNIT-RECORD
USAGE
USAGE CLAUSE
USE
UT-S-PRINTER
UT-S-READER
V
VALVE
VALUE CLAUSE
VERB
W
WORK AREA

WORKING-STORAGE
WORKING-STORAGE SECTION
WORK SPACE
WRITE
X
XX
Y
YES-BRANCH
Z
ZERO
ZEUS
01
77
9
0011 DIAGNOSTIC CODE, INVALID
0102 DIAGNOSTIC CODE, INVALID
0103 DIAGNOSTIC CODE, INVALID
0105 DIAGNOSTIC CODE, INVALID
0120 DIAGNOSTIC CODE, INVALID
0391 DIAGNOSTIC CODE, INVALID
0431 DIAGNOSTIC CODE, INVALID
0521 DIAGNOSTIC CODE, INVALID
0701 DIAGNOSTIC CODE, INVALID
0791 DIAGNOSTIC CODE, INVALID
0831 DIAGNOSTIC CODE, INVALID
0871 DIAGNOSTIC CODE, INVALID
1001 LEVEL NUMBER
1002 HEADER,SECTION
1003 PARAGRAPH-NAME
1004 WORD, INVALID
1005 ORDER, INVALID
1006 DECLARATIVES SECTION
1007 SPACE
1008 PAREN, SPACE BEFORE
1009 COPY
1010 PAREN, SPACE AFTER
1011 RECORDING MODE
1012 FILE-NAME, UNIQUE
1014 FILE TAPE CLAUSE, MULTIPLE
1015 EXTERNAL-NAME
1016 CLAUSE, MULTIPLE
1017 CLAUSE, INVALID ENTRY
1023 FILE-NAME, INVALID
1024 FILE-NAME, UNDEFINED
1028 SYNTAX, SENTENCE
1029 FILE-NAME, UNDEFINED
1030 WORD, INVALID
1036 DATA-NAME, INVALID
1037 DATA DESCRIPTION
1038 DATA-NAME, LEVEL
1039 DATA-NAME QUALIFIER
1040 LEVEL NUMBER 77
1041 ENTRY, INVALID
1042 ASSIGN CLAUSE
1043 PERIOD, END
1044 WORD, INVALID
1045 ORDER, INVALID
1053 CLAUSE, INVALID
1055 FILE-NAME, INVALID
1056 FILE-NAME, SELECT
1060 REPORT WRITER STATEMENT
1061 CLAUSE, DUPLICATE
1067 HEADER, INVALID
1077 CONTINUATION MARGIN
1078 CONTINUATION CHARACTER
1079 ALPHANUMERIC LIT
1080 PERIOD, SPACE BEFORE
1081 PERIOD, SPACE AFTER
1083 CHARACTER, ILLEGAL
1084 COMMA, SPACE BEFORE
1085 PICTURE SIZE
1086 A-MARGIN
1087 B-MARGIN
1090 DELETE THRU
1091 MARGIN, PROC-NM
1095 HEADER, INCOMPLETE
1097 PROGRAM-ID
1098 CONTINUATION CHARACTERS
1099 ITEM, INVALID
1100 SEQUENCE
1101 NEXT PAGE OPTION
1110 SUM CLAUSE
1120 COMMA, SPACE AFTER
1128 HEADER, ED
1129 HEADER, ID
1130 HEADER
1140 LITERAL, NUMERIC > 32768
1141 SYSTEM-NAME, INVALID
1143 USE STATEMENT
1150 SAME AREA CLAUSE
1155 SYSTEM-NAME, INVALID
1157 SYSTEM-NAME, MISSING EXT-N
1188 DIAGNOSTIC CODE, INVALID
1201 DIAGNOSTIC CODE, INVALID
1210 DIAGNOSTIC CODE, INVALID
1234 DIAGNOSTIC CODE, INVALID
1301 DIAGNOSTIC CODE, INVALID
1404 DIAGNOSTIC CODE, INVALID
1511 DIAGNOSTIC CODE, INVALID
1530 DIAGNOSTIC CODE, INVALID
1550 DIAGNOSTIC CODE, INVALID
1880 DIAGNOSTIC CODE, INVALID
2010 REDEFINES CLAUSE
2029 LEVEL NUMBER 01

2030	LEVEL NUMBER 77
2034	PICTURE; GROUP ITEM
2039	PICTURE CONFIGURATION
2049	OPEN STATEMENT
2050	RECORD DISTRIBUTION
2051	DATA-NAME, TRACK-AREA
2075	PICTURE, NUMERIC
2107	USAGE CLAUSE
2110	WRITE-ONLY CLAUSE
2128	PICTURE, VALUE CLAUSE
2133	LABEL RECORD CLAUSE
2146	RECORD CONTAINS CLAUSE
2149	VALUE CLAUSE, OCCURS
2151	PICTURE, ELEMENTARY ITEM
2160	RENAMES CLAUSE
2183	LEVEL 01 ENTRY
2202	DIAGNOSTIC CODE, INVALID
2237	DIAGNOSTIC CODE, INVALID
2242	DIAGNOSTIC CODE, INVALID
2301	DIAGNOSTIC CODE, INVALID
2371	DIAGNOSTIC CODE, INVALID
3001	DATA-NAME, NOT DEFINED
3002	DATA-NAME, NOT UNIQUE
3003	DATA-NAME, NOT DEFINED
3004	DATA-NAME, NOT UNIQUE
3005	DATA-NAME, NOT VALID
3006	DATA-NAME, NOT DEFINED
3009	IDENTIFIER-2
3010	TO, SUPERFLUOUS
3011	IDENTIFIER-1
3014	COMPILER ERROR
3021	DATA-ITEM, DUPLICATED
3456	DIAGNOSTIC CODE, INVALID
3731	DIAGNOSTIC CODE, INVALID
4001	CONDITION
4002	STATEMENT, INCOMPLETE
4003	STATEMENT, MISSING
4004	ITEM, ILLEGAL
4007	ITEM, MISPLACED OR MISSING
4008	ITEM, SUPERFLUOUS
4010	PAREN, LEFT
4011	CHARACTER, ILLEGAL
4012	LITERAL COMPARISON
4013	IF STATEMENT, INCOMPLETE
4019	ARITHMETIC OPERAND
4020	SIGN, ILLEGAL
4024	GIVING
4030	CONDITION, INCOMPLETE
4032	IF STATEMENT
4041	OPEN STATEMENT, INCOMPLETE
4042	FILE-NAME, MISSING
4050	SYNTAX
4052	OPERAND
4056	SYNTAX
4057	SORT KEYS
4070	AT END
4075	NEXT SENTENCE
4076	SUBSCRIPTING
4085	RECORD DESCRIPTION
4086	CONDITION, ILLEGAL
4091	SYNTAX, OPERAND
4096	LOGIC
4098	ALTER STATEMENT
4140	DIAGNOSTIC CODE, INVALID
5004	PICTURE SIZE
5005	COMPILER ERROR
5006	COMPILER ERROR
6001	PROCESSING ERROR
6321	DIAGNOSTIC CODE, INVALID
6754	DIAGNOSTIC CODE, INVALID

Appendix H

CONTENTS OF THE STUDENT REFERENCE MANUAL

CONTENTS

- Foreword 2
- Acknowledgement 3
- Introduction to Computer-Administered Instruction 7
- COBOL Programming Course 8
- COURSE FEATURES 10
  - How to Use the Manual 11
  - Student Stations 12
  - Proctor Assistance 13
  - Checking In and Out of the Course 14
  - Answering Questions 15
  - How to Request Glossary Information 16
  - How to Comment on the CRT 17
  - REVIEW, PRACTICE, QUIZ and RECAP Options 18
  - CAI Course Organization 19
- CRT PROCEDURES 20
  - CRT Display Terminal 21
  - CRT Keyboard: Alphanumeric Keys 22
  - CRT Keyboard: Carriage Keys 23
  - CRT Keyboard: Carriage Keys (Special) 24
  - CRT Keyboard: Mode Keys 25
  - How to Make a Correction on the CRT 26
  - How to Use the Light Pen 27
  - Moving the Cursor 28
  - CRT Character Set 29
- COBOL PROGRAMMING 30
  - COBOL: What It Is and Does 31
  - COBOL Reserved Words (A-L) 32
  - COBOL Reserved Words (M-Z) 33
  - Rules for Data-Names, File-Names, and Record Names 34
  - Rules for Procedure-Names 35
  - Other Names 36
  - Rules for Using COBOL Formats 37
  - Formats for Program A 38
  - Formats for Program B 39

CONTENTS *(Continued)*

COBOL PROGRAM DIVISIONS 40
- Identification Division 41
- Environment Division 42
- Data Division for Programs C and D 43
- Simple File Description 44
- Record Descriptions 45
- PICTURE Clauses 45A
- Record Layouts and Program Specifications 46
- Working-Storage Section 47
- Procedure Division: Programs A and B 48
- Procedure Division: Programs C and D 49
- Counters 50
- Moving: Steps A and B 51
- Moving Data by COBOL and Computer 52
- Moving Records Part by Part 53
- Garbage 54

FLOWCHARTING 55
- Flowcharting Summary 56
- Flowcharting Symbols 57
- Flowcharting Supplies 58
- Flowcharting Documentation 59

USING COMPUTER FACILITIES 60
- Rules for Using the COBOL Coding Form 61
- Coding Form Checklist 62
- Compile-And-Go Checklist 63
- How to Punch a Card 64
- Preparation for Keypunching 65
- Job Control Language 66
- Debugging a Computer Program 67
- Diagnostics 68
- Sequence Numbers 69

SUPPLEMENTARY COURSE INFORMATION 70
- Appendix A A - 1
- Appendix B B - 1
- Appendix C C - 1
- Appendix D D - 1
- Appendix E E - 1

APPENDIX A A - 1
- About This Course A - 2
- Patterns and Names A - 3
- Examples of Procedures A - 4
- Topic AEA Directions: Situation 1 A - 5
- Topic AEA Directions: Situation 2 A - 6
- Jail Paragraph A - 7
- Go To Work A - 8
- Judgment A - 9
- Pantry-Restock A - 10

CONTENTS *(Continued)*

APPENDIX A *(Continued)*

- Getalist A - 11
- Topic AGA Directions A - 12
- Topic AHA Directions A - 13
- Program A Specifications A - 14
- Betting Instructions A - 15
- Program A Specifications (Complete) A - 16

APPENDIX B B - 1
- Data Division Structure B - 2
- Flowchart 1 B - 3
- Flowchart 2 B - 4
- Flowchart 3 B - 5
- Program B Specifications B - 6

APPENDIX C C - 1
- Sample Record Descriptions C - 2
- Program C Specifications and Record Layouts C - 3
- Sample Input Record for Program C C - 4
- Input Records 1 - 4 C - 5
- Program C Specifications C - 6

APPENDIX D D - 1
- Which Printout Will Result? D - 2
- Program D Specifications D - 3
- Sample Program Specifications D - 4
- Student Progress D - 5
- Record Layouts - Student Progress D - 6
- Test Data for Checking Flowchart D - 7
- Truck-Parts D - 8
- Record Layouts - Truck Parts D - 9
- Marigolds D - 10
- Record Layouts - Marigolds D - 11
- Format for IF Statements D - 12
- Flowcharting a Problem D - 13
- Program D Messages D - 14
- Voter Eligibility D - 15
- Literals D - 16
- The DISPLAY Statement D - 17
- Coding Multi-Line Statements D - 18

APPENDIX E E - 1
- Division A Topic Descriptions E - 2
- Division B Topic Descriptions E - 4
- Division C Topic Descriptions E - 6
- Division D Topic Descriptions E - 7

Appendix I

SUPER-EDITOR

COURSE NAME: SUPRED

SUPER-EDITOR is an on-line support program for authors and developers of instructional content. It is called a course because the elements are stored in the instructional text files and the logic is part of the instructional logic files. As far as the operating system is concerned, the user "signs on" to SUPRED as a student. In reality, the program is an authoring aid with no instructional pretenses.

The role of SUPER-EDITOR in the courseware subsystem is to facilitate the manipulation of text files for naive as well as sophisticated authors and editors when either authoring or editing is done interactively on-line. The SUPER-EDITOR program was created in COURSEWRITER III to implement the commands available in the Editor portion of the executive program (ZEUS) so that the user does not have to know the Editor command set. An author is required to know only the operational definitions of the SUPER-EDITOR modes, as explained in Table I-1.

Table I-1

SUPER-EDITOR Modes

Symbol	Mode	Operation
0	ANTIDOTE	Retrieves a deleted element if no other operation has intervened between DELETE and ANTIDOTE.
1	DELETE	Inactivates a disc location.
2	DISPLAY	Retrieves the contents of a disc location on the CRT screen for viewing, but does not permit any permanent alteration of the element.
3	CREATE	Activates a disc location and permits the user to store text on the element.
4	MODIFY	Accesses an active disc location and permits the user to change the content and/or structure of the resident element.
5	PRE-FORMAT	Copies a previously stored template into a newly activated disc location and permits the user to input content into the structure.
6	DISC DISPLAY	Retrieves the disc location number of an element being viewed via DISPLAY and places it in the upper left corner of the element.
7	EDITOR	Allows the user to write his own Editor commands, two at a time. Adds these to the Editor page.
8	SIGN OFF	Allows the user to exit from SUPRED.
9	DISC TITLES	Executes the list command which shows the label of all specified active disc locations and the name of the last person who worked on each.
A	JOB ENTRY	Allows the user to retrieve and execute any program in the D-file.
B	TITLE MODIFY	Allows the user to make changes in the label of a disc location and to change the content and/or structure of the element, if desired.

Mode selection is done on-line with a master element (Figure I-1), by filling in the information requested under "please specify".

Mode Element

```
                         SUPER-EDITOR MODES:

      0. ANTIDOTE              5. PRE-FORMAT              A. JOB ENTRY
      1. DELETE                6. DISC DISPLAY            B. TITLE MODIFY
      2. DISPLAY               7. EDITOR
      3. CREATE                8. SIGN OFF
      4. MODIFY                9. DISC TITLES
  PLEASE SPECIFY:
                FIRST          LAST          CURSOR BLOCK
  MODE          DISC           DISC           (2 DIGITS)

    .           .....          .....              ..

                                                        PRESS SEND.
```

Figure I-1

The user must replace the dots in the mode element (lower portion of Figure I-1) with the relevant information for each position.

User positions on the mode element are explained below:

MODE is a one-position field that specifies the mode desired. The user replaces the single dot with the proper symbol (0-9), A, B) shown on the mode element (Figure I-1).

FIRST DISC is a five-position field that specifies the first in a possible range of disc locations to which the given mode is to be applied. The user replaces the first dot with the file designation (D, F, Q) and the remaining dots with the disc location number.

LAST DISC is similar to the previous field, but specifies the last location of the range. If only one element is to be accessed, this entry is the same as the entry for FIRST DISC.

CURSOR BLOCK is a two-position field that specifies the number of the block where the cursor will appear when the element is displayed on the CRT screen.

PRESS SEND is a direction that tells the user to enter his input data into the computer. With the Sanders terminal, the user performs this operation by depressing the SEND BLOCK key.

The following example indicates that the user wished to apply the DISPLAY mode to every disc location in the D-file from 4519 to 4582, inclusive, and position the cursor in block 2 for each of these elements:

MODE	FIRST DISC	LAST DISC	CURSOR BLOCK (2 DIGITS)
2	D4519	D4582	02

If a user specifies a LAST-DISC location which is numerically lower than the stated FIRST-DISC location, the chosen mode will be applied only to the FIRST-DISC location. If the block designation is higher than the actual number of blocks in the display, the cursor will appear in the last character position on the screen (referred to as "end of memory").

SUPRED is most useful when large portions of the text files are to be manipulated because the author can create, review, and modify an instructional package without going back and forth between "student-mode" and "author-mode." This review, of course, does not preclude the necessity to debug the instructional logic, but it can save time and effort by correcting errors that might cause further problems.

Appendix J

VOYEUR

COURSE NAME: VOYEUR

VOYEUR is a course only in the sense that the text is in the instructional text files and the logic is in the instructional logic files. It is really a support program to aid proctors and course managers in performing their duties.

In Project IMPACT, the VOYEUR program was used on a proctor terminal to monitor the progress of a group of on-line students. The four main elements of the program are described below:

(1) Student Status. Lists each active student by number, with the following additional information: (a) location in the course (by course label), (b) most recent quiz score (as CTR 29 =), and (c) last input to the system (as Buffer 0 =). This information is retrieved from each student's line data area and updated every time the proctor executes a PRESS SEND on the monitor terminal.

Figure J-1 shows a portion of VOYEUR output on student status. Retrieving information from each student's line data area, the program can report on 10 students at a time.

Portion of VOYEUR Output

```
03/26/73    16:26


STUDENT =          255       LABEL = BIA16A       CTR 29 = 0
BUFFERO=SIGN OFF

STUDENT =          282       LABEL = BHA00A       CTR 29 = 177
BUFFERO=LINK B
```

Figure J-1

(2) Trouble Report. Presents a form to be completed by the proctor after each interaction with a student. As shown in Figure J-2, the information to be supplied by the proctor includes the following:

CWIP—course location where interaction occurred.
STUDENT NO.—identification of student needing help.
CUBICLE—student station where interaction occurred.
PROC.—initials of the proctor answering the call.
OM—initials of the chief proctor of the shift.
TROUBLE—alphanumeric code of the possible problems presented in four categories.
OTHER—explanation of the trouble code "50".

VOYEUR Element for Trouble Reports

```
CWIP...    STUDENT NO. ...   CUBICLE..   PROC. ...   OM ...    ID.    1008   DATE   9 JAN 1973    TR 14
                                                                              TIME:   18:57

                                      TROUBLE ..       XX IN QUIZ

          IP                       CW                     HRDWRE                   MISC

   A1-TYPO CRT            G1-RESPONSE ANAL.        M1-LOST CUE COUNT      S1-LOCKED KBD

   A2-TYPO P-SCOPE        G2-CURSOR POS. NG        M2- P-SCOPE NG         S2- P-S OFF CENTER

   A3-TYPO MANUAL         G3-CRT SCRAMBLED         M3-SYSTEM CRASH        S3-MULT. CURSORS

   A4-CRT TEST NG         G4- P-SCOPE CUE NG       M4-LIGHT PEN NG        S4-MISSING MATLS

   A5-PSCOPE TEST NG      G5-WRONG BRANCH          M5-CRT NG              S5-IDM PROBLEM

   A6-MANUAL TEST NG      G6-CRT BLANKED           M6-LINE TROUBLE        S6-S ERROR

   A7-DIRECTIONS UNCLR    G7-WRONG FDBK            M7-BULB BURN-OUT       S7-CMI CUB BUSY

   A8-SUBJ MATTER ?       G8-WRONG POINTS          M8-VIEWLEX NG          S8-IP INTERNAL

   A9-XX IN P SECTION     G9-TP I/O ERROR

   B1-OPTION PROB.        H1-EXCESS PROC.

   B2-GLOSSARY PROB

   B3-PROGRAM ?

   50-OTHER....................................
```

Figure J-2

Additional blank lines are also provided for more explanation if desired. The identification number (ID.) of the trouble report is supplied by the system, along with the date, the time, and the code of the student group (this is TR12 in Figure J-2).

In Project IMPACT, copies of these trouble reports were referred to appropriate staff specialists for solving or eliminating the problems. This Referee Action was also added to the trouble report on-line. All data entered by proctors, referees, and the system were incorporated into a data base for output by variables such as number of trouble reports per student, per course label, per type of problem, etc.

(3) System Status. Pre-formatted command of the executive program (Zeus) that allows a proctor to inspect the status of the system, including the latest messages on the console, the size of the partitions, and the relative activity of each partition in the main computer.

(4) Display. Pre-formatted command of the executive program (ZEUS) that allows a proctor to quickly look at any element in the course.

The proctor can move easily from one element to another in VOYEUR by entering on the CRT screen a one-character designator that initiates an immediate branch to the element desired. These designators are:

V for Student Status,
T for Trouble Report,
S for System Status
Y for Display.